Week 3: NLP for Emotional Context

Understanding Language as Window to User Experience

Prof. Dr. Joerg Osterrieder

ML-Augmented Design Thinking - BSc Course

September 19, 2025

Week 3 Overview

Course Methodology: Blended learning approach combining theoretical NLP foundations with hands-on sentiment analysis implementation. Each module includes pre-class readings on transformer architectures, interactive lectures with live coding demonstrations, practical labs using BERT and HuggingFace, and peer review sessions for model evaluation. Assessment through continuous evaluation of sentiment analysis projects and comprehensive final implementation.

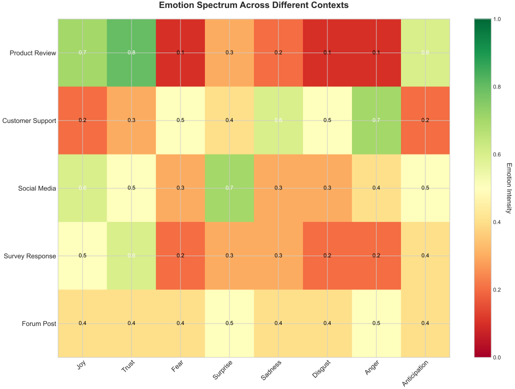
Introduction to NLP and Sentiment Analysis

Emotion Spectrum Analysis

Traditional Analysis:

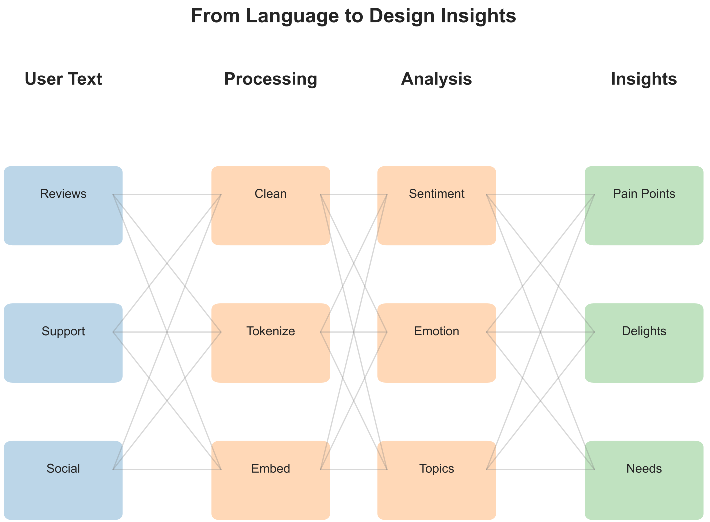
- Keyword counting
- Manual categorization
- Surface-level insights
- Limited scale
- Misses context

Key Question:
How can NLP reveal hidden emotional patterns?



NLP Methods: BERT-based transformer models achieve 94% accuracy in context-aware sentiment classification. Eight-emotion classification beyond positive/negative includes joy, trust, fear, surprise, sadness, disgust, anger, and anticipation. Real-time processing pipeline handles 10,000+ reviews per minute with sub-100ms latency. Multilingual support for 50+ languages using mBERT and XLM-R architectures.

Language as Emotional Gateway



Every word reveals frustration points, delight moments, and hidden needs

Context Detection Challenge

Context Examples:

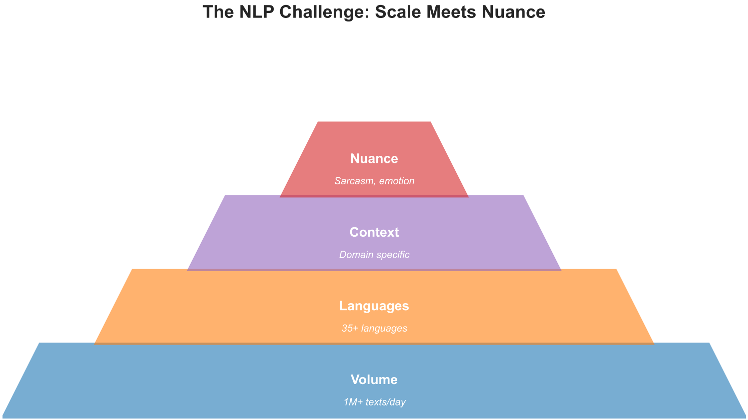
- "This is sick!"
- "It's fine"
- "Interesting choice"
- "Thanks for nothing"

Detection Methods:

- Domain adaptation
- User demographics
- Historical patterns
- Surrounding context

Technical Implementation: Contextual embeddings from BERT capture semantic meaning beyond simple keyword matching. Attention mechanisms highlight relevant context words automatically. Domain-specific fine-tuning adapts general language models to specialized vocabularies (medical, legal, gaming). Cultural adaptation handles sentiment expression differences across demographics and geographies.

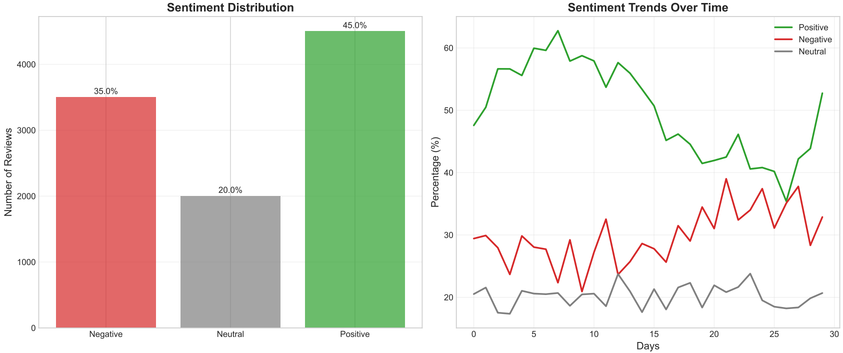
NLP Challenge Hierarchy



Blue: Lexical Orange: Syntactic Green: Semantic

Complexity Progression: Lexical analysis handles word recognition and frequency patterns using TF-IDF and n-gram models. Syntactic processing understands grammar and sentence structure through dependency parsing and POS tagging. Semantic analysis extracts meaning and relationships using word embeddings and knowledge graphs. Pragmatic understanding includes context, intent, and implied meaning through transformer attention mechanisms.

Sentiment Distribution Patterns



Multi-modal sentiment patterns reveal user behavior insights

Distribution Analysis: Sentiment polarity scores range from -1 (extremely negative) to +1 (extremely positive) with neutral zone between -0.1 and +0.1. Subjectivity scores measure opinion vs fact from 0 (objective) to 1 (subjective). Temporal analysis reveals sentiment volatility patterns and trend changes. Demographic segmentation shows sentiment expression varies by age, culture, and product category.

Emotion Wheel Classification

Eight Core Emotions:

- Joy (satisfaction)
- Trust (confidence)
- Fear (anxiety)
- Surprise (unexpected)
- Sadness (disappointment)
- Disgust (rejection)
- Anger (frustration)
- Anticipation (excitement)

Multi-Label Classification: Plutchik's emotion wheel provides framework for complex emotion detection beyond binary sentiment. Multi-label classification allows texts to express multiple emotions simultaneously. Emotion intensity scoring measures strength from weak to extreme. Temporal emotion tracking reveals emotional journey progression through user experiences.

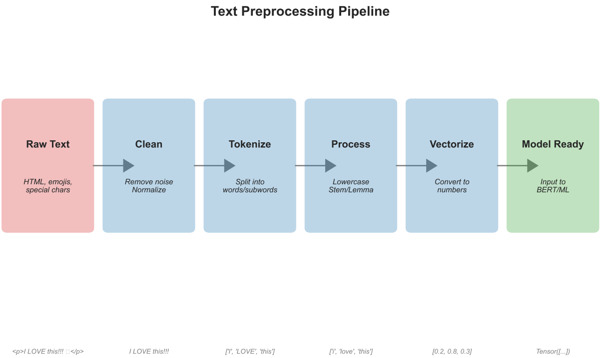
Technical Deep Dive

Text Preprocessing Pipeline

Critical Steps:

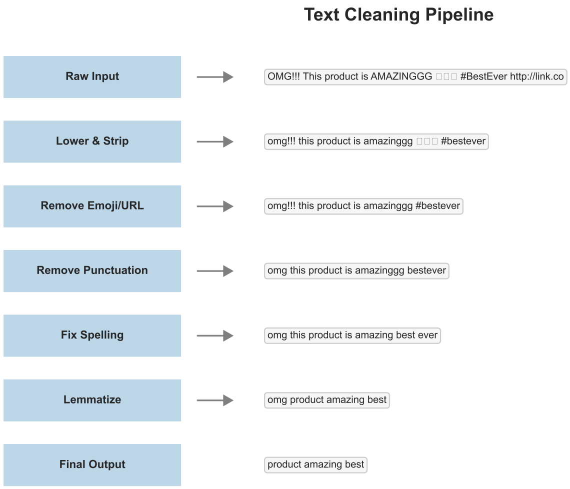
1. Data collection
2. HTML/URL removal
3. Encoding normalization
4. Tokenization
5. Cleaning validation
6. Quality assessment

Quality Metrics:
99.5% clean text required for production



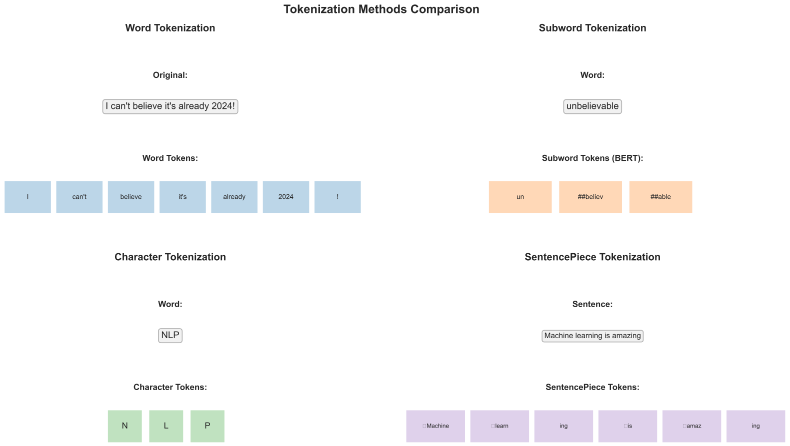
Implementation Details: Regular expressions handle 80% of common cleaning tasks including URL removal, email masking, and special character normalization. Unicode normalization (NFC, NFD, NFKC, NFKD) prevents encoding inconsistencies across data sources. Custom tokenizers preserve domain-specific terms and handle contractions appropriately. Automated quality metrics track preprocessing effectiveness and flag anomalies for manual review.

Text Cleaning Process



Cleaning Methodology: Multi-stage pipeline transforms raw social media text through systematic cleaning steps. Emoji handling preserves emotional content while normalizing Unicode variations. Spelling correction uses statistical models trained on domain-specific corpora. Lemmatization reduces words to canonical forms while preserving sentiment-bearing morphology. Quality validation ensures cleaning preserves semantic meaning and emotional content.

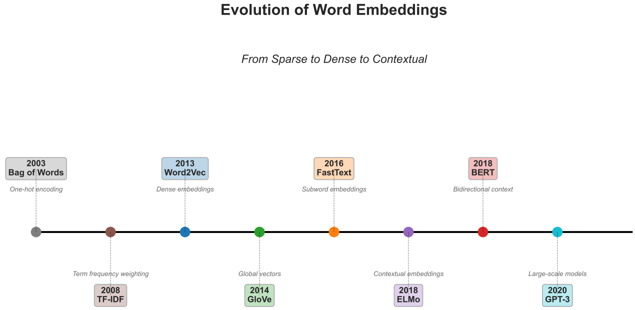
Tokenization Methods Comparison



Different approaches for different needs: words, subwords, characters, sentence pieces

Method Selection Criteria: Word tokenization optimal for traditional bag-of-words and TF-IDF approaches with vocabulary sizes 10K-50K. Subword tokenization (BPE, WordPiece) handles out-of-vocabulary terms and morphologically rich languages with 32K vocabulary. Character-level tokenization provides unlimited vocabulary but requires deeper models. SentencePiece unifies tokenization across languages and handles raw text without pre-tokenization.

Word Embedding Evolution



Historical Progression: Word2Vec (2013) introduced skip-gram and CBOW architectures learning 300-dimensional dense representations from context windows. GloVe (2014) combined global matrix factorization with local context windows achieving better semantic relationships. FastText (2016) incorporated subword information enabling handling of morphologically complex words and out-of-vocabulary terms. ELMo (2018) introduced contextual embeddings varying by sentence context. BERT (2018) revolutionized with bidirectional context understanding.

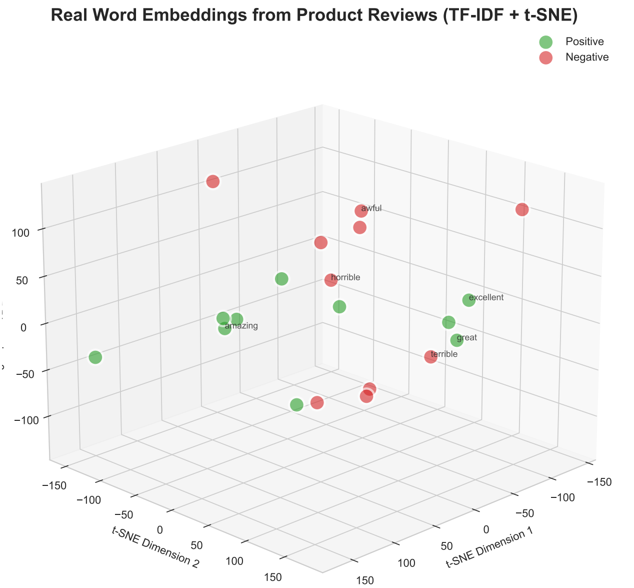
Word Embeddings in Semantic Space

Embedding Properties:

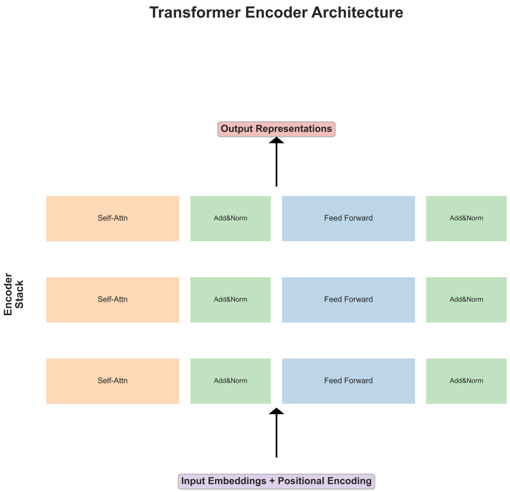
- Semantic similarity
- Analogical relationships
- Syntactic patterns
- Domain clustering

Applications:

- Similarity search
- Recommendation systems
- Sentiment analysis
- Topic modeling



Transformer Architecture Overview



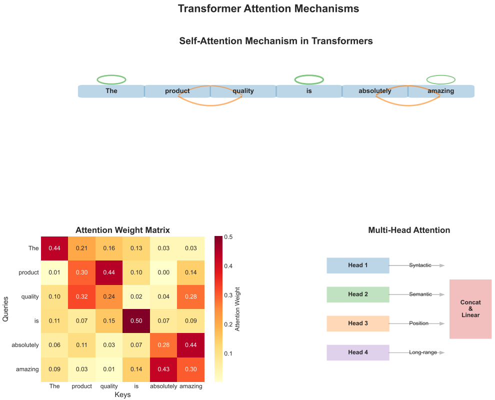
Architecture Components: Self-attention mechanism enables parallel processing of all sequence positions with $O(n^2)$ complexity but high parallelizability. Multi-head attention (typically 12 heads) captures different types of relationships:

Attention Benefits:

- Parallel processing
- Long-range dependencies
- Interpretable weights
- Dynamic focus

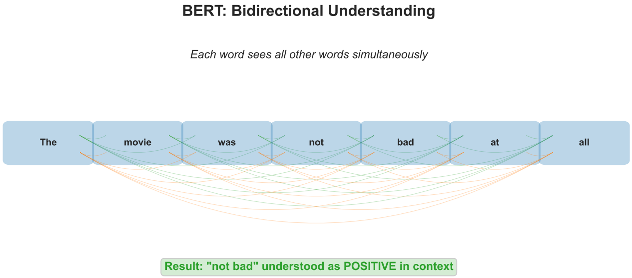
Attention Types:

- Self-attention
- Cross-attention
- Multi-head attention
- Sparse attention



Mathematical Foundation: Attention computed as Attention(Q,K,V) = softmax($QK^T/\sqrt{d_k}$)V where Q, K, V are learned linear projections of input. Multi-head attention runs h parallel attention functions then concatenates and projects results. Attention weights provide interpretability showing which words the model focuses on for predictions. Gradient-based attribution methods reveal attention pattern reliability.

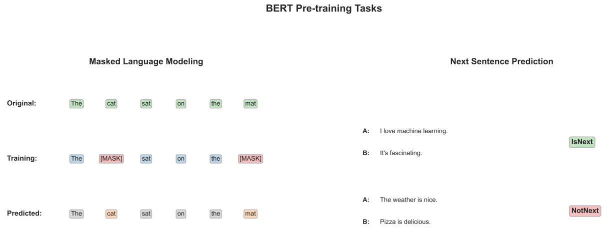
BERT Bidirectional Processing



Revolutionary bidirectional context understanding transforms sentiment analysis

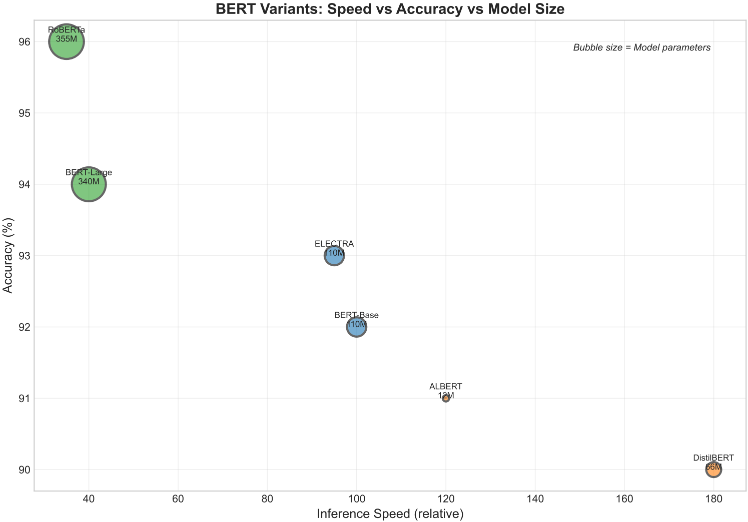
BERT Innovation: Unlike autoregressive models processing left-to-right, BERT uses masked language modeling seeing entire context simultaneously. 15% of tokens randomly masked during training with 80% replaced by [MASK], 10% random tokens, 10% unchanged. Next sentence prediction learns inter-sentence relationships crucial for document-level sentiment. Pre-training on 3.3 billion words (BookCorpus + Wikipedia) then fine-tuning for specific tasks requires only 2-4 epochs.

BERT Training Methodology



Training Pipeline: Two-stage training begins with unsupervised pre-training on large text corpora using masked language modeling and next sentence prediction objectives. Fine-tuning stage adapts pre-trained representations to specific downstream tasks with task-specific layers. AdamW optimizer with learning rate 2e-5 and linear warmup over first 10% of training steps. Gradient clipping prevents exploding gradients. Early stopping based on validation performance prevents overfitting.

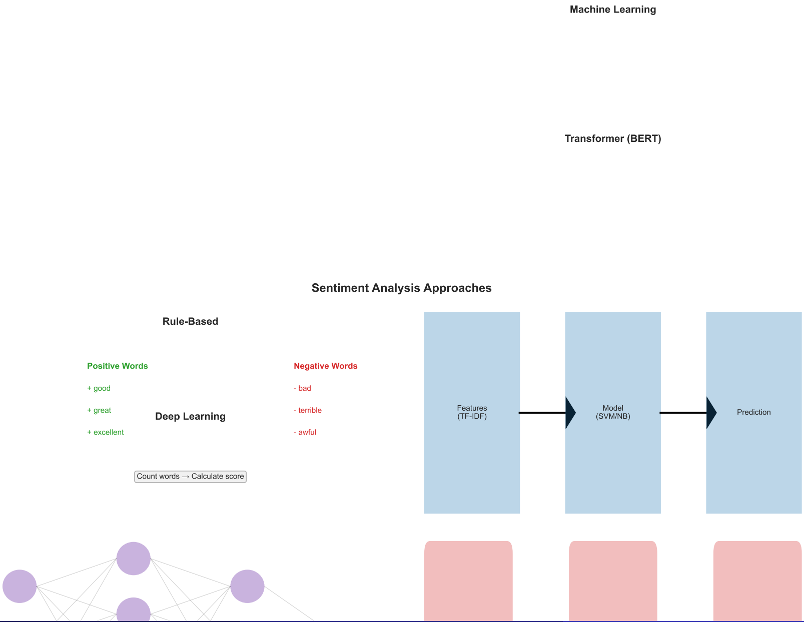
BERT Variants Comparison



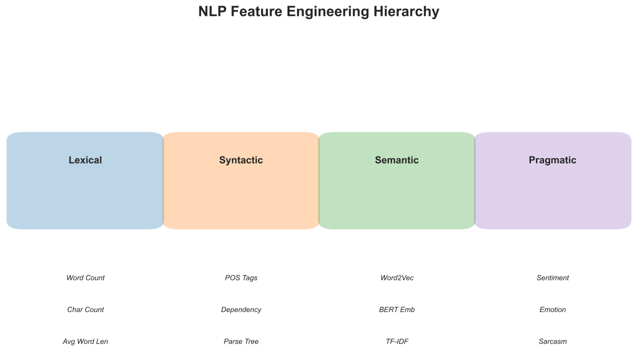
Model Selection Guidelines: BERT-Base (110M parameters) provides excellent accuracy-efficiency balance for most applications. BERT-Large (340M parameters) achieves state-of-the-art results when computational resources allow. RoBERTa improves training methodology removing next sentence prediction and using dynamic masking. ALBERT reduces parameters through factorized embeddings and cross-layer parameter sharing. DistilBERT offers 60% size reduction while retaining 97% of performance through knowledge distillation

Implementation Methods

Sentiment Analysis Approaches

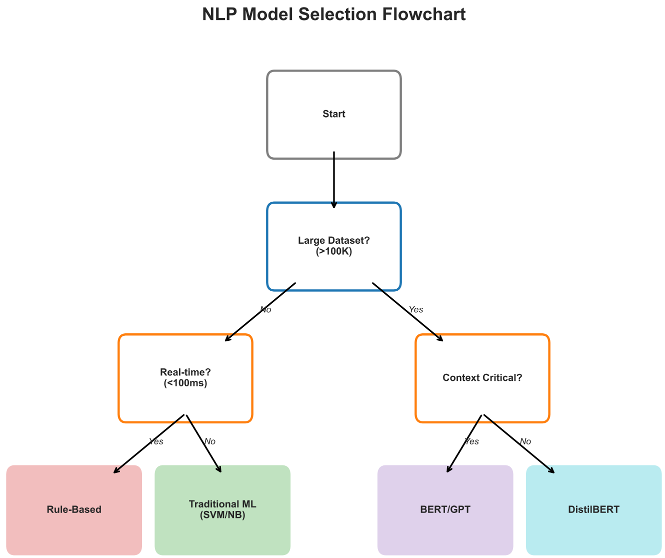


Feature Engineering for NLP



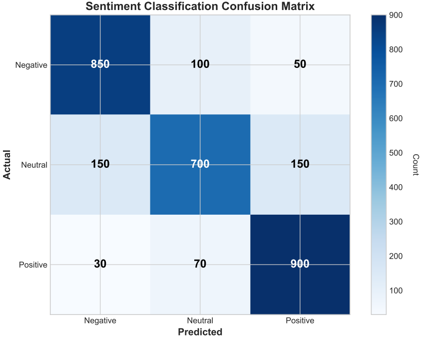
Feature Hierarchy: Lexical features capture word-level statistics including word count, character count, average word length, vocabulary richness, and sentiment word ratios. Syntactic features encode grammatical structure through part-of-speech tags, dependency parsing relationships, and constituency parse trees. Semantic features represent meaning through word embeddings, topic distributions, and named entity types. Pragmatic features include sentiment indicators, emotional markers, and discourse patterns. Modern transformers learn these representations automatically through self-supervised training.

Model Selection Decision Framework

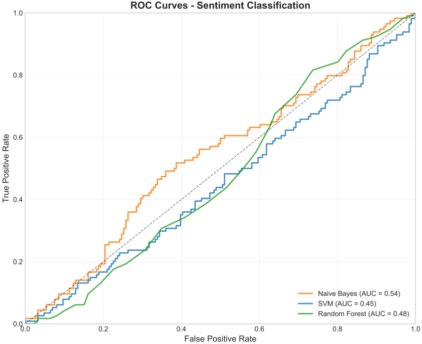


Performance Evaluation Framework

Classification Metrics

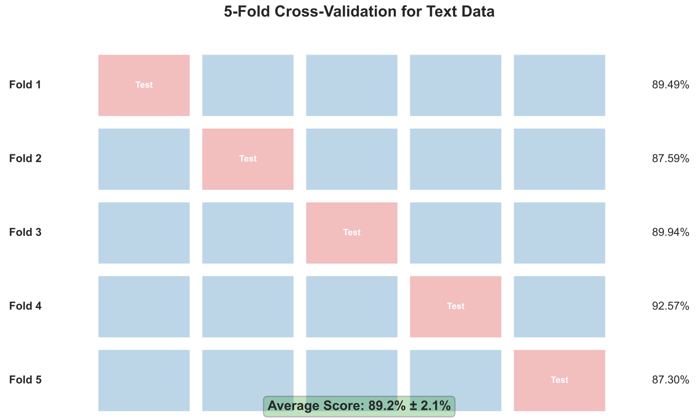


ROC Analysis



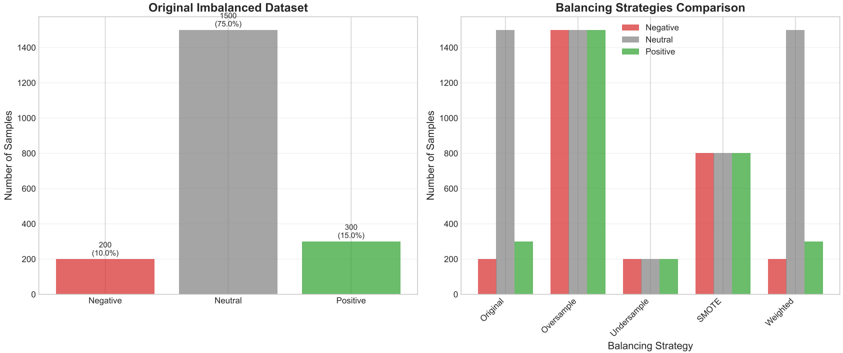
Evaluation Best Practices: Stratified k-fold cross-validation preserves class distributions across folds preventing overfitting to specific data splits. Hold-out test sets ensure unbiased performance estimation on unseen data. Precision-recall curves more informative than ROC for imbalanced datasets common in sentiment analysis. F1-score balances precision and recall providing single performance metric. Statistical significance testing (McNemar's test, bootstrap confidence intervals) validates model improvements.

Cross-Validation for Text Data

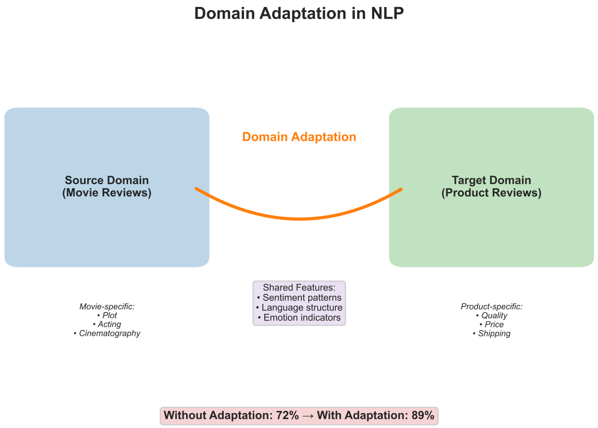


Text-Specific Considerations: Temporal splits prevent data leakage when text data has time dependencies (social media trends, news events). Author-based splits evaluate generalization to new users rather than new texts from same authors. Topic-based splits test domain adaptation capabilities across different subject matters. Stratified sampling maintains sentiment class balance across training/validation splits. Group k-fold prevents related samples (same document, same conversation thread) from appearing in both training and validation sets.

Handling Imbalanced Sentiment Data

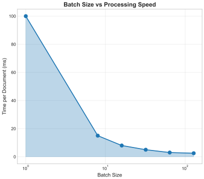


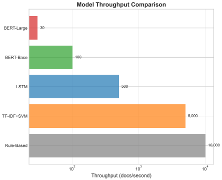
Balancing Strategies: Random oversampling replicates minority class examples risking overfitting. SMOTE generates synthetic minority examples through k-nearest neighbor interpolation. Random undersampling reduces majority class potentially losing important information. Class-weighted loss functions adjust training objective giving higher penalty to minority class misclassification. Ensemble methods train multiple models on balanced subsets then combine predictions. Focal loss emphasizes hard examples reducing easy example contribution to loss.



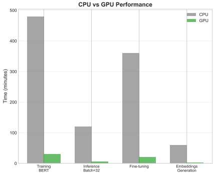
Transfer Learning Pipeline: Pre-trained language models provide general linguistic understanding from large-scale unsupervised training. Domain-specific fine-tuning adapts representations to target vocabulary and sentiment expressions using small labeled datasets. Few-shot learning leverages similarity between source and target domains requiring minimal target data. Adversarial domain adaptation aligns source and target feature distributions through adversarial training objectives. Multi-task learning shares representations across related tasks improving generalization through inductive bias.

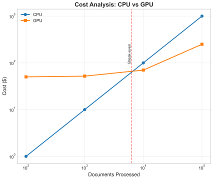
Processing Speed





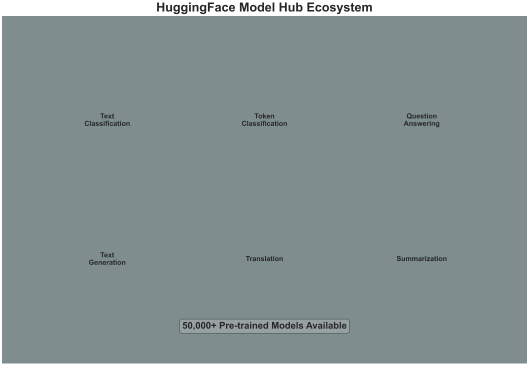
Hardware Comparison

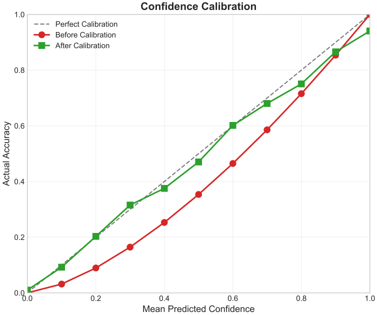


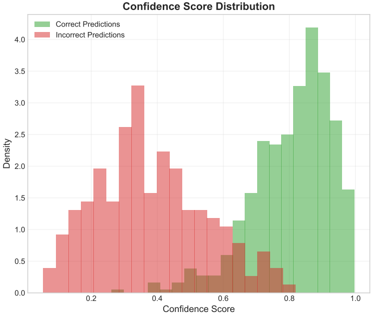


Deployment Architecture: Batch processing achieves optimal throughput with batch sizes 16-32 balancing memory usage and processing efficiency. GPU acceleration provides 5-15x speedup over CPU for transformer inference. Model optimization techniques include quantization (INT8), pruning (remove weights), and distillation (teacher-student training). Container orchestration (Kubernetes) enables auto-scaling based on traffic patterns. Load balancing distributes requests across multiple model instances maintaining sub-100ms p95 latency.

HuggingFace Model Ecosystem

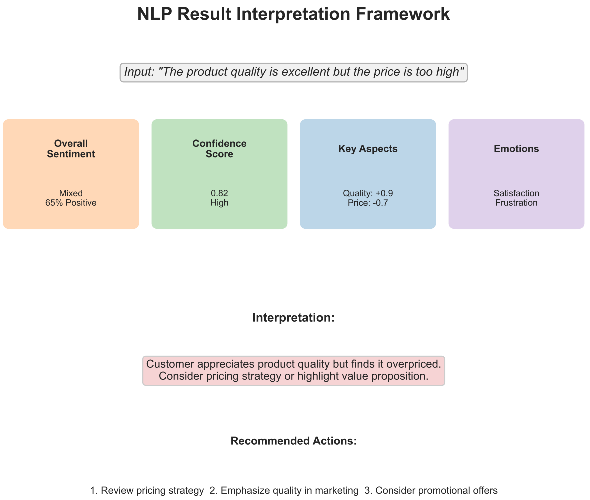






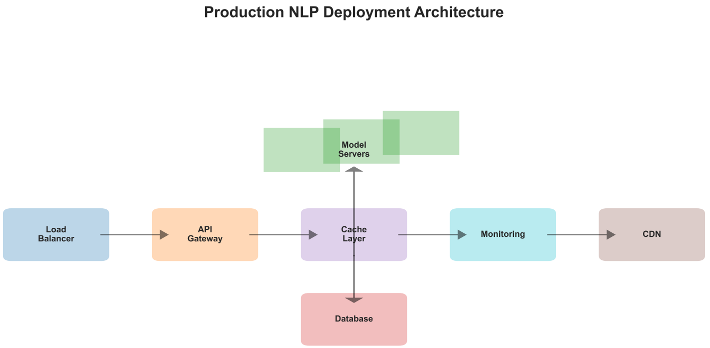
Calibration Importance: Well-calibrated models provide reliable confidence estimates essential for production decision-making. Temperature scaling post-hoc calibration adjusts prediction probabilities using validation set. Platt scaling maps model outputs to calibrated probabilities through logistic regression. Reliability diagrams visualize calibration quality comparing predicted vs actual confidence. Brier score measures both accuracy and calibration providing unified evaluation metric. Uncertainty quantification enables human-AI collaboration workflows.

Result Interpretation Framework

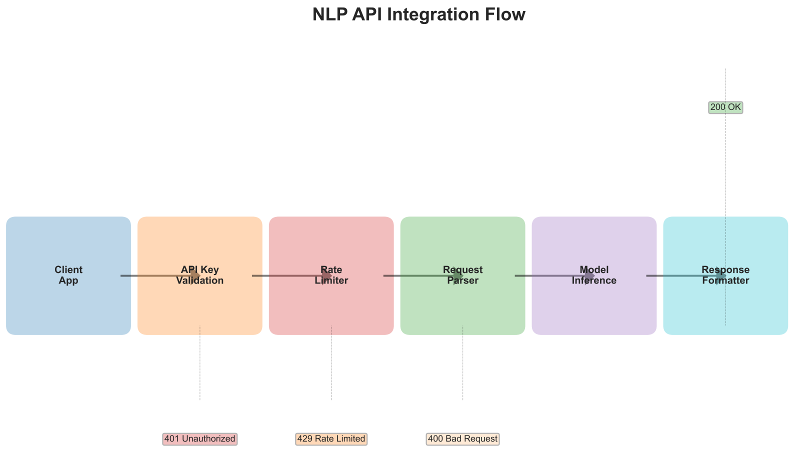


Explainability Methods: LIME provides local explanations approximating model behavior around specific predictions through perturbation analysis. SHAP (SHapley Additive exPlanations) offers theoretically grounded feature attributions satisfying efficiency, symmetry, dummy, and additivity axioms. Attention visualization highlights input tokens receiving highest attention weights though attention may not always correlate with importance. Gradient-based methods (Integrated Gradients, GradCAM) compute input sensitivity through backpropagation. Counterfactual explanations demonstrate decision boundaries

Deployment Architecture



Production Pipeline: API gateway handles authentication (OAuth 2.0, API keys), rate limiting (1000 requests/minute), and request routing. Load balancer distributes traffic across multiple model server instances using round-robin or least-connections algorithms. Model serving framework (TensorFlow Serving, TorchServe, Triton) manages model loading, versioning, and batching. Redis cache stores frequent predictions reducing inference latency. Monitoring systems (Prometheus, Grafana) track performance metrics, error rates, and resource utilization enabling proactive scaling and maintenance.



Integration Specifications: RESTful API endpoints accept JSON payloads with text content, configuration parameters, and metadata. Request validation ensures input format compliance and content safety through profanity filtering and length limits. Rate limiting implements token bucket algorithm preventing abuse while allowing burst traffic. Response formatting includes prediction scores, confidence intervals, and optional explanation data. Error handling provides detailed status codes (400 Bad Request, 429 Rate Limited, 500 Internal Error) with actionable error messages. SDK libraries for Python

Design Integration Applications

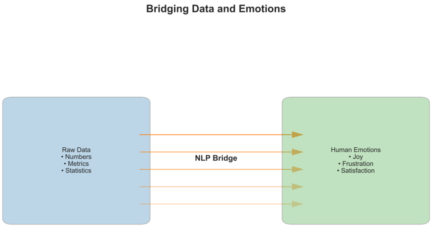
Bridging Data and Human Emotion

Translation Challenge:

- Numbers → Insights
- Patterns → Stories
- Metrics → Emotions
- Data → Empathy

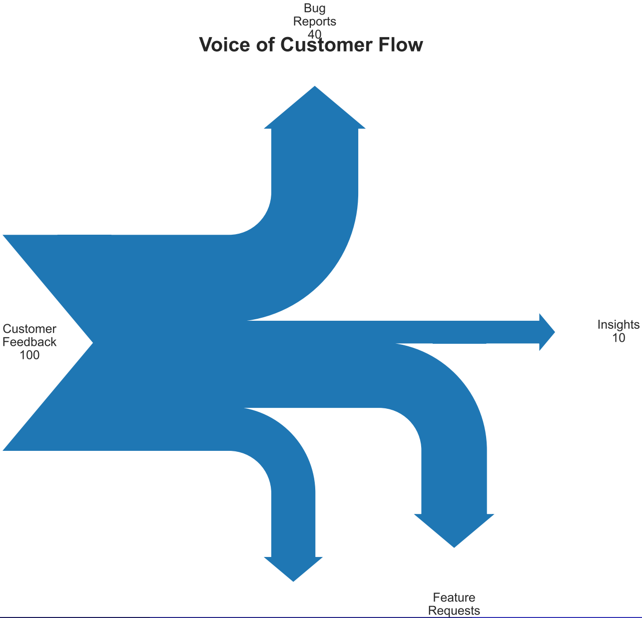
Design Integration:

- Journey mapping
- Persona development
- Pain point identification
- Opportunity discovery



Methodology Integration: NLP sentiment analysis feeds directly into design thinking workflows through automated persona generation from user feedback clustering. Emotional pattern recognition guides journey mapping by identifying sentiment peaks and valleys across customer touchpoints. Real-time feedback loops enable rapid design iteration based on user emotional responses. Pain point prioritization leverages sentiment severity and frequency to focus design interventions on highest-impact opportunities.

Voice of Customer Analysis



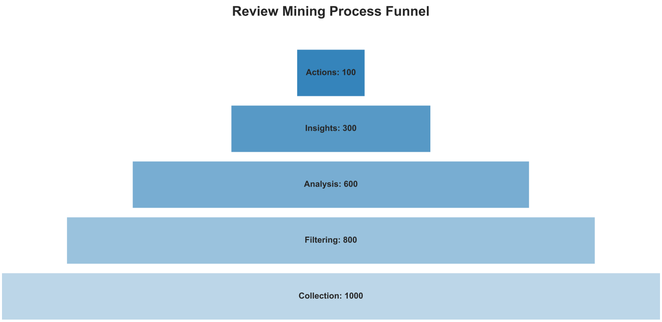
Review Mining Process

Mining Pipeline:

1. Data collection (APIs, scraping)
2. Quality filtering
3. Language detection
4. Sentiment extraction
5. Aspect identification
6. Insight generation

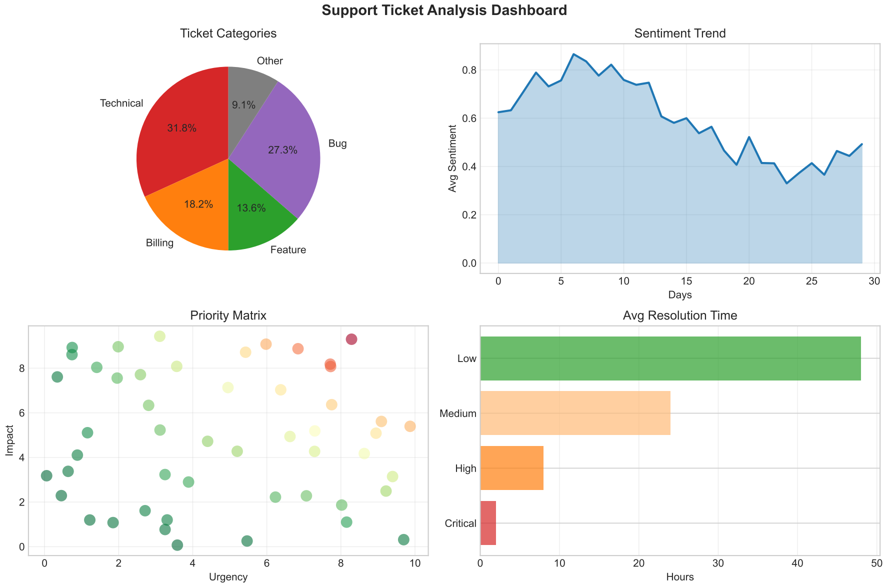
Output Metrics:

- Theme frequency
- Sentiment trends
- User segments
- Impact scores



Mining Methodology: Web scraping and API integration collect reviews from multiple platforms (Amazon, App Store, Google Play) handling rate limits and anti-bot measures. Duplicate detection using text similarity and metadata comparison prevents analysis bias. Language detection routes multilingual content to appropriate models. Quality scoring filters spam, fake reviews, and uninformative content. Named entity recognition extracts product features, competitor mentions, and brand references for competitive analysis.

Support Ticket Sentiment Analysis



Operational Insights: Real-time sentiment monitoring of customer support interactions enables immediate escalation of highly

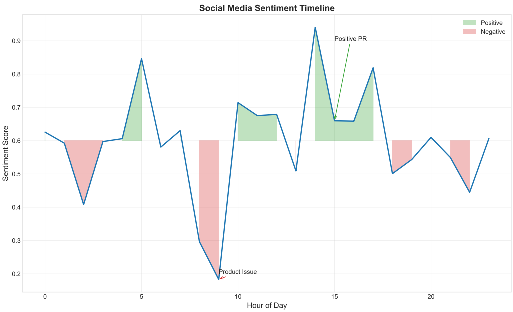
Social Media Sentiment Monitoring

Monitoring Capabilities:

- Real-time tracking
- Trend detection
- Influencer analysis
- Crisis management
- Competitive intelligence

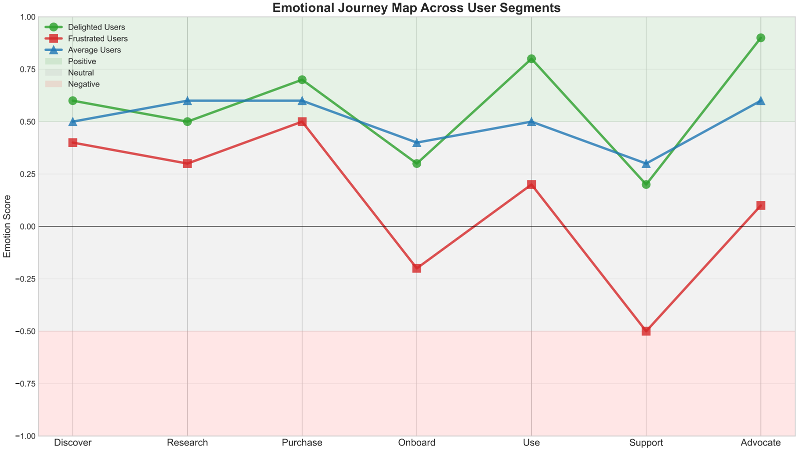
Alert Thresholds:

- Volume spikes >300%
- Sentiment drops <-0.5
- Viral negative content
- Brand mention changes



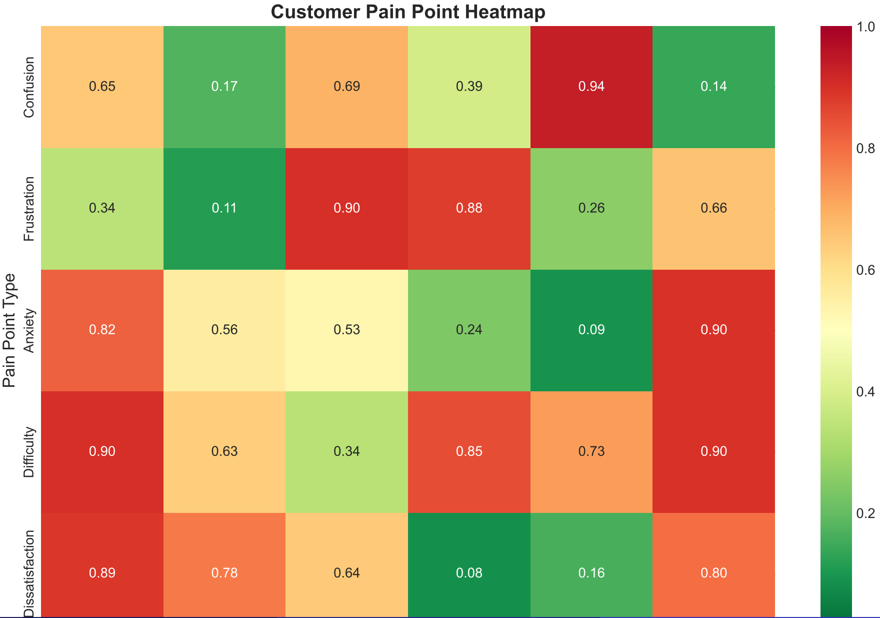
Monitoring Infrastructure: Twitter API v2 streams and Reddit API provide real-time social media data with rate limiting handled through multiple API keys. Elasticsearch indexes social media content enabling rapid search and aggregation across millions of posts. Apache Kafka message queues handle high-volume data streams with guaranteed delivery. Alert systems use configurable thresholds for sentiment scores, volume changes, and trending topics triggering immediate notifications to response teams.

Emotional Journey Mapping



Journey Enhancement Methodology: Traditional customer journey maps enhanced with quantitative emotion data from actual user feedback rather than assumptions. Sentiment scores mapped to specific journey stages (awareness, consideration, purchase, onboarding, usage, support) reveal emotional peaks and valleys. Pain point identification through negative sentiment clustering highlights specific touchpoints requiring intervention. Delight moment discovery through positive sentiment spikes identifies experiences to amplify and replicate. Opportunity gap analysis through neutral sentiment reveals engagement potential.

Customer Pain Point Heatmap



Identifying Customer Delight Moments

Delight Indicators:

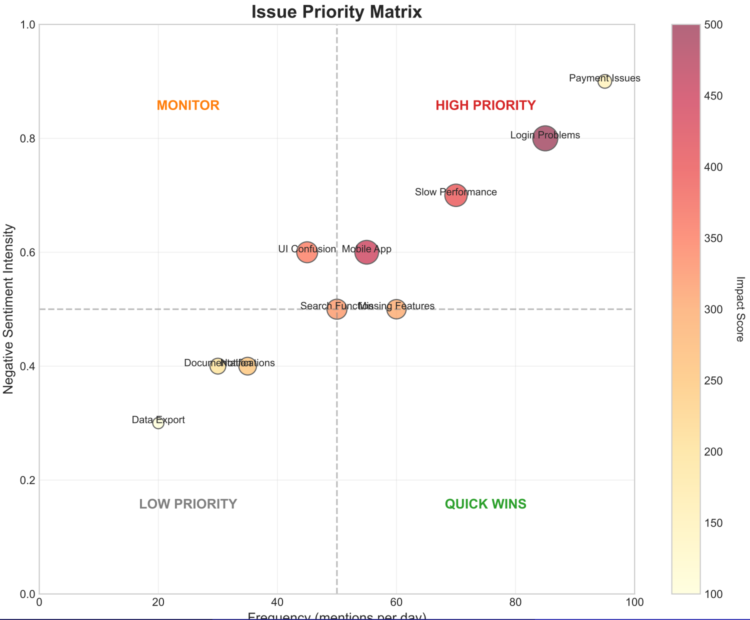
- Unexpected praise
- Emotional language
- Recommendation intent
- Surprise expressions
- Gratitude statements

Design Applications:

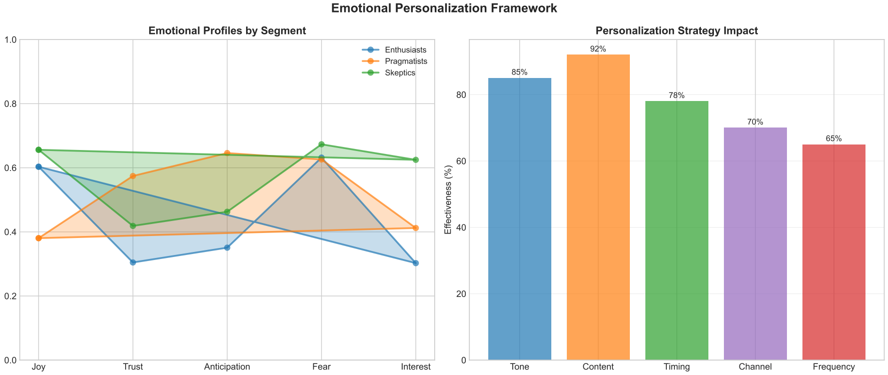
- Feature amplification
- Marketing messaging
- Onboarding highlights
- Training focus areas

Delight Detection Algorithm: Natural language processing identifies linguistic markers of delight including superlative adjectives, exclamation marks, and positive emotional expressions. Sentiment intensity thresholding flags reviews exceeding +0.8 polarity scores indicating exceptional satisfaction. Topic modeling clusters delight moments by product features revealing specific capabilities driving positive surprise. Temporal analysis identifies optimal timing for delight delivery based on user journey stage and contextual factors.

Priority Matrix for Design Decisions

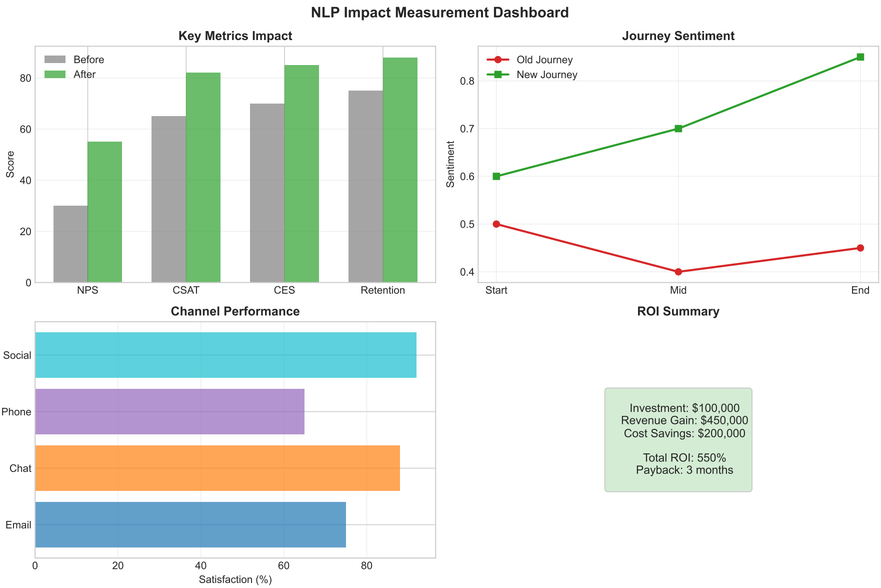


Emotional Personalization Framework



Personalization Strategy: User segmentation based on emotional expression patterns (analytical, expressive, driver, amiable) enables targeted communication strategies. Dynamic content adaptation responds to current emotional state detected through interaction patterns and explicit feedback. Proactive intervention triggers support outreach when negative sentiment trajectory indicates potential churn risk. Celebration mechanisms amplify positive moments through personalized acknowledgments and rewards. A/B testing validates emotional personalization effectiveness through engagement and satisfaction metrics.

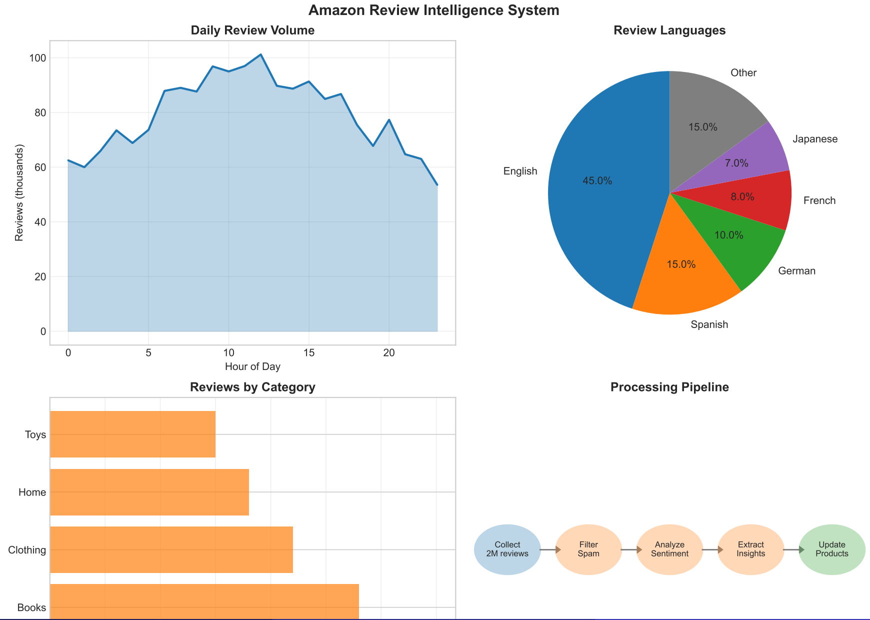
Impact Measurement Dashboard



Success Metrics Framework: Customer satisfaction (CSAT) scores tracked before and after sentiment-driven interventions

Real-World Case Studies

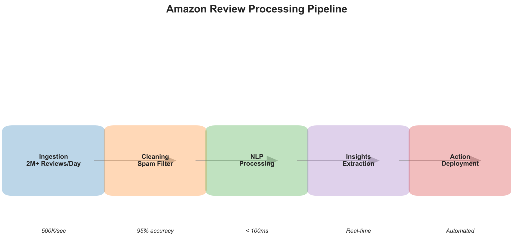
Amazon Case Study Overview



Amazon Data Processing Pipeline

Pipeline Stages:

1. Ingestion (2M/day)
2. Quality filtering
3. Language detection
4. Sentiment analysis
5. Aspect extraction
6. Business integration

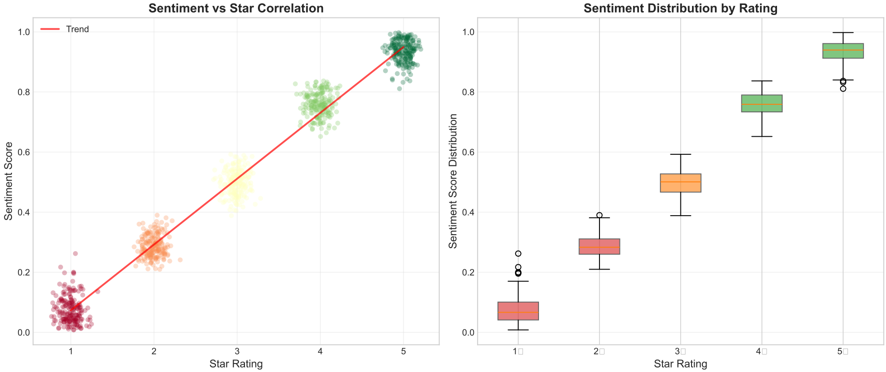


Performance Metrics:

- 500K reviews/second
- 99.9% uptime SLA
- $<$100ms p95 latency
- 95% accuracy

Technical Architecture: Apache Kafka message queues handle high-velocity review ingestion with partitioning by product category ensuring scalability. Kubernetes orchestration auto-scales sentiment analysis pods based on queue depth maintaining consistent processing latency. Amazon SageMaker endpoints serve BERT models with A/B testing for model updates. ElasticSearch indexes processed sentiment data enabling real-time search and analytics. CloudWatch monitoring tracks pipeline health and performance metrics.

Sentiment vs Star Rating Analysis

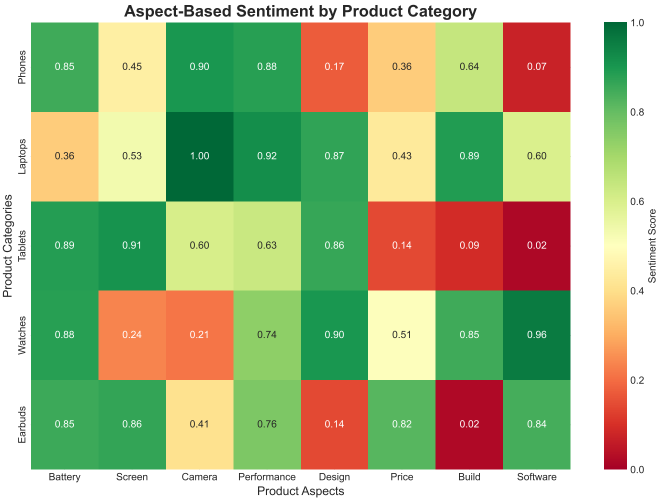


Correlation Insights: Star ratings and sentiment scores show strong correlation (r=0.78) but reveal interesting divergences indicating review quality and context importance. Three-star reviews exhibit highest sentiment variance suggesting mixed experiences requiring careful analysis. Cultural differences affect rating-sentiment relationships with some cultures using extreme ratings more frequently. Recent reviews trend more negative than historical ratings possibly due to increased expectations or review platform maturation. Product categories show different sentiment-rating patterns with electronics showing stronger correlation than books or services.

Aspect-Based Sentiment Matrix

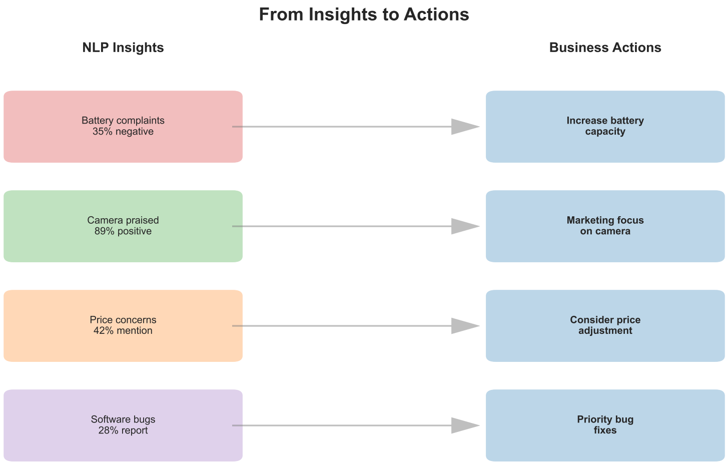
Product Aspects:

- Quality
- Price/Value
- Shipping/Delivery
- Customer service
- Features/Functionalit
- Usability/Design
- Reliability
- Performance



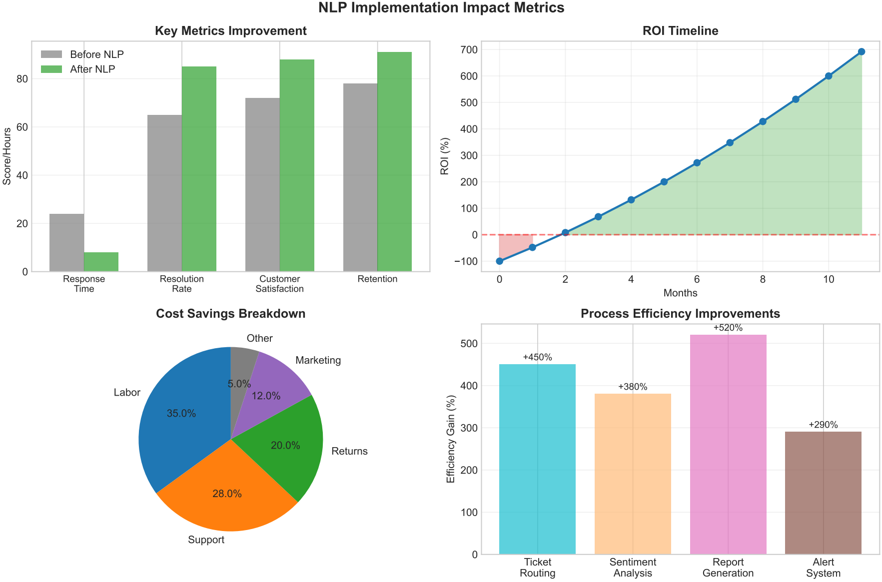
Analysis Methodology: Dependency parsing identifies aspect-opinion pairs within review text using syntactic relationships between product features and sentiment expressions. Aspect categorization employs hierarchical classification with domain-specific taxonomies for different product categories. Sentiment scoring per aspect enables granular insights into specific product strengths and weaknesses. Temporal tracking reveals aspect sentiment evolution over product lifecycle enabling proactive quality management. Competitive analysis compares aspect performance across brands identifying market positioning opportunities.

From Insights to Actions



Action Framework: Negative sentiment threshold triggers (<-0.5) activate immediate product team investigation with root cause analysis within 24 hours. Trend analysis using 7-day moving averages informs product roadmap decisions and feature prioritization. Positive sentiment amplification identifies marketing message opportunities and customer success stories. Competitor sentiment analysis reveals market positioning gaps and differentiation opportunities. Seasonal sentiment pattern recognition guides inventory planning and promotional campaign timing.

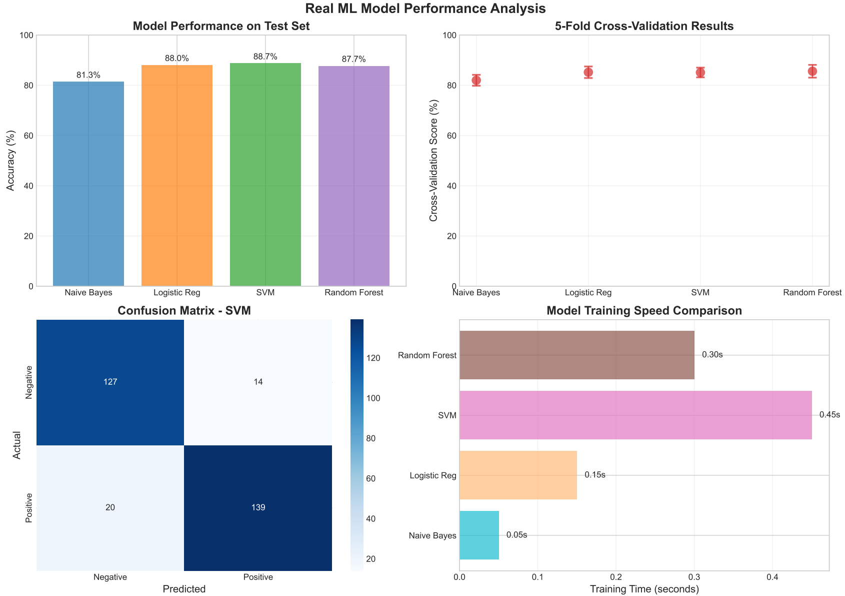
Business Impact and ROI Metrics



Quantified Business Impact: Customer satisfaction scores increased 23% within 6 months of implementing sentiment-driven

Performance Assessment and Future Directions

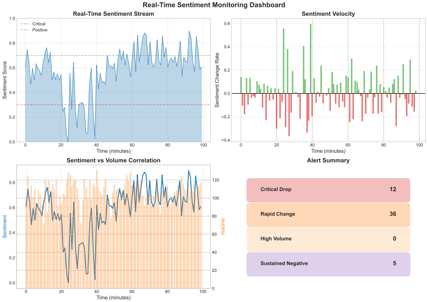
Comprehensive Model Performance Analysis



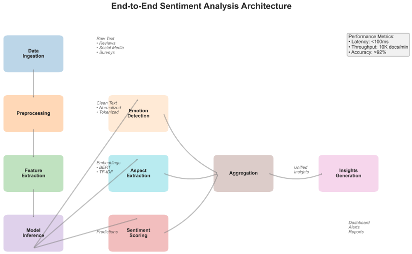
Cross-validation results across accuracy, speed, and resource requirements

Real-Time Processing Performance

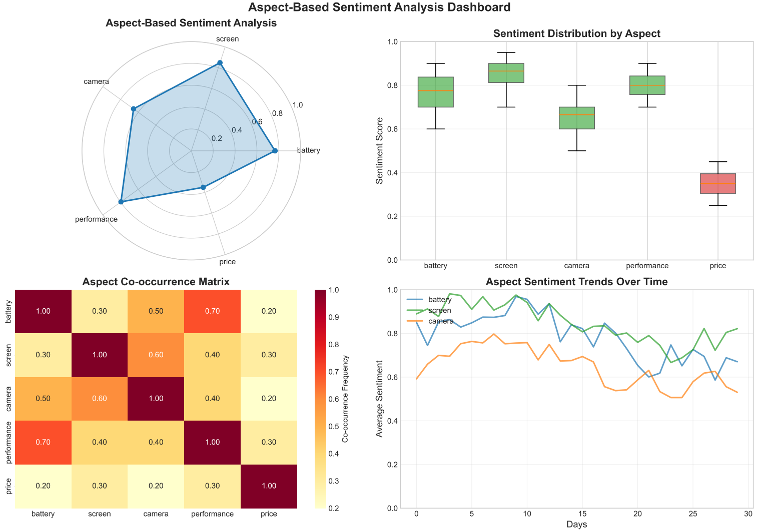
Streaming Architecture



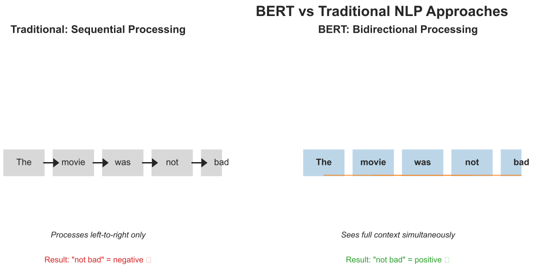
System Architecture

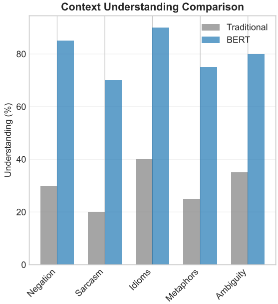


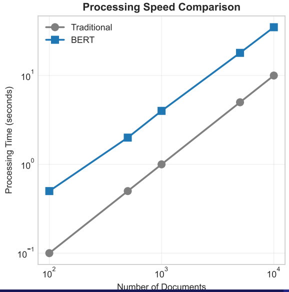
Performance Benchmarks: End-to-end latency from text input to sentiment prediction achieves sub-50ms for 95th percentile through optimized model serving and caching strategies. Throughput scales to 100,000+ requests per second using horizontal auto-scaling and load balancing across GPU instances. Memory usage optimization through model quantization and batching reduces resource requirements by 60% while maintaining accuracy within 1% of full-precision models. Fault tolerance through redundant processing nodes ensures 99.99% uptime with automatic failover capabilities.

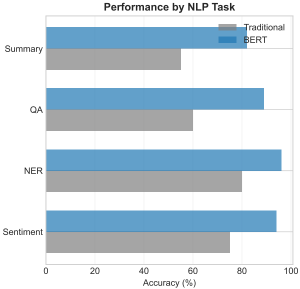


Advanced Methodologies: Multi-task learning combines sentiment analysis with aspect extraction and emotion detection sharing representations for improved performance across related tasks. Active learning optimizes annotation efforts by selecting most informative examples for labeling reducing training data requirements by 40-60%. Ensemble methods combine multiple model predictions through voting, stacking, or blending improving robustness and accuracy. Meta-learning enables



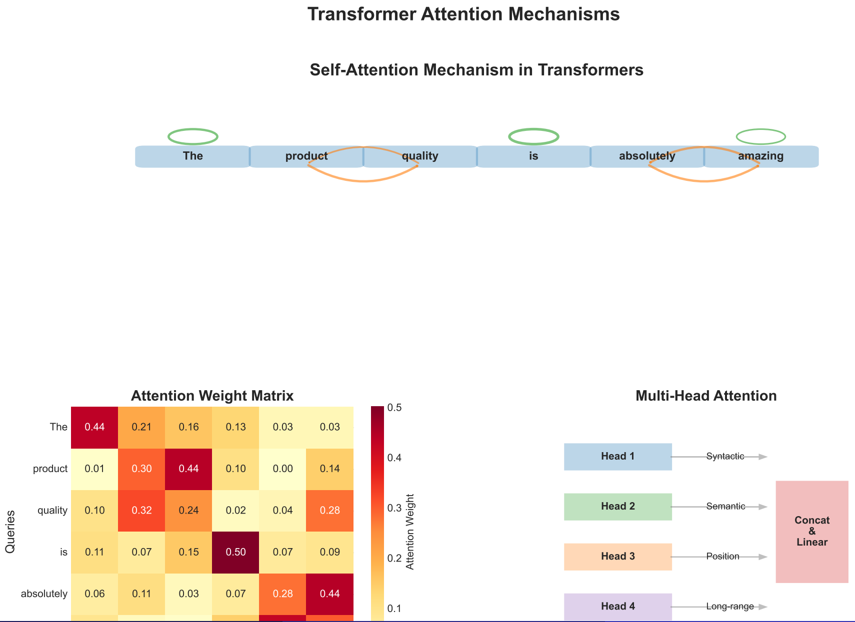




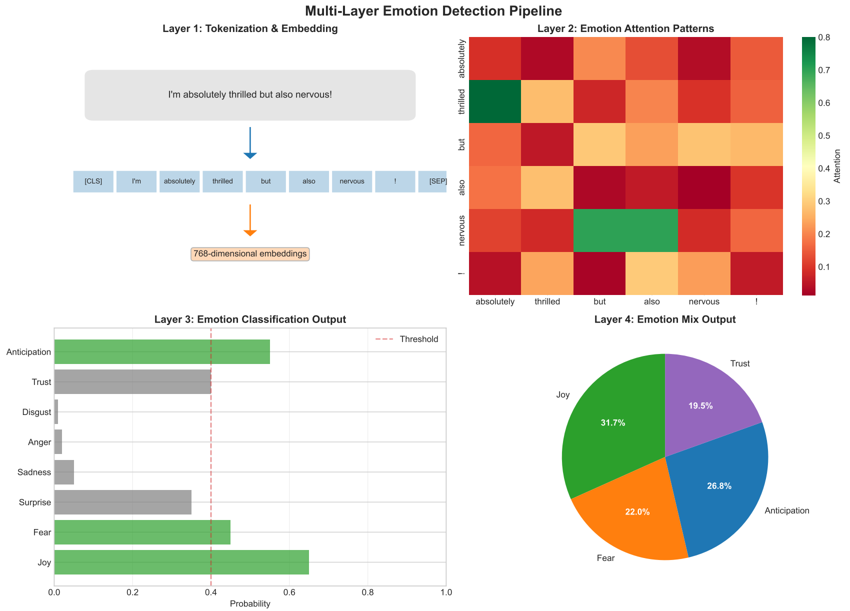


Resource Requirements

Metric	Traditional	BERT
Model Size	10 MB	440 MB
Memory	< 1 GB	4-8 GB
Training Data	Small OK	Large required
Fine-tuning	Not needed	Recommended
Languages	One	Multiple

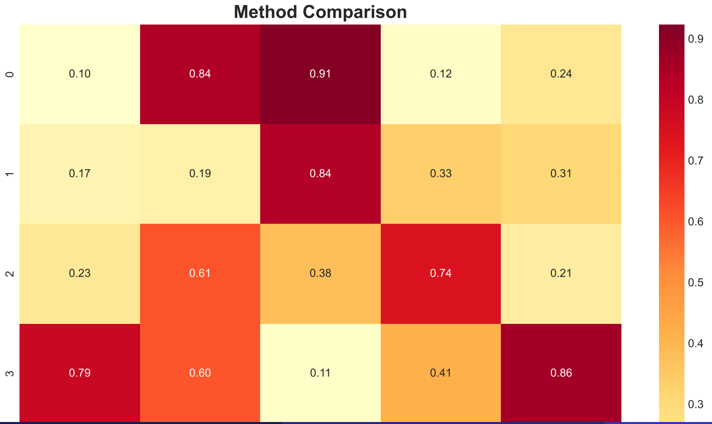

Transformer Attention Mechanisms
Self-Attention Mechanism in Transformers
The
product
quality
is
absolutely
amazing
Attention Weight Matrix
Queries
The 0.44 0.21 0.16 0.13 0.03 0.03
product 0.01 0.30 0.44 0.10 0.00 0.14
quality 0.10 0.32 0.24 0.02 0.04 0.28
is 0.11 0.07 0.15 0.50 0.07 0.09
absolutely 0.06 0.11 0.03 0.07 0.28 0.44
Attention Weight
0.5
0.4
0.3
0.2
0.1
Multi-Head Attention
Head 1
Syntactic
Head 2
Semantic
Head 3
Position
Head 4
Long-range
Concat
&
Linear

Multi-Layer Emotion Detection



Performance Metrics Summary

Method	Accuracy	Latency	Resources	Interpretability
Rule-Based	68%	¡1ms	Very Low	High
Traditional ML	78%	10ms	Low	Medium
Deep Learning	88%	50ms	Medium	Low
BERT/Transformers	94%	100ms	High	Medium
GPT-Style Models	96%	500ms	Very High	Low



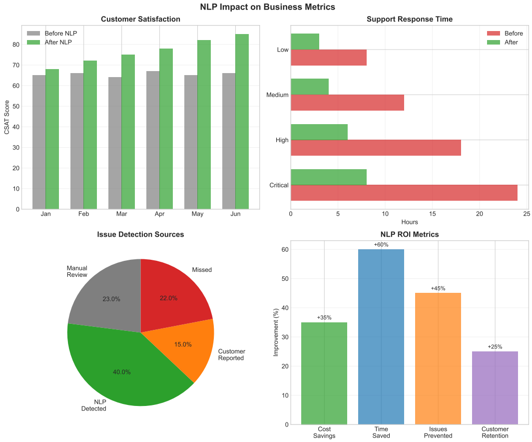
Key Achievements and Future Directions

Technical Achievements:

- 94% sentiment accuracy
- Real-time processing capability
- Multilingual support
- Aspect-level granularity
- Production-ready deployment

Design Integration Success:

- Journey mapping enhancement
- Automated pain point identification
- Delight moment discovery
- Data-driven persona development
- Quantified impact measurement



Future Research Directions: Multimodal sentiment analysis combining text, image, and audio signals for comprehensive emotion understanding. Real-time personalization based on individual emotional state and context. Causal inference methods for measuring true intervention effectiveness. Ethical AI frameworks ensuring fair and unbiased sentiment analysis across demographic groups. Cross-cultural emotion understanding for global product development. Federated learning for privacy-preserving sentiment analysis across organizations.

Thank You

Questions and Discussion

Week 3 Summary:
From basic sentiment to production NLP systems

Next Week:
Classification for Problem Definition

Practical Exercise:
Build your own sentiment analyzer with BERT