

## Lexicon-Based Sentiment Analysis

### Core Idea

Use pre-defined dictionary with word sentiment scores

### Example Sentence

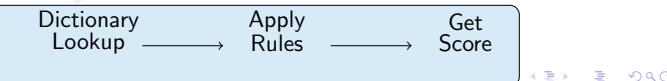
"The movie was good, but the ending was terrible!"

### Dictionary Lookup

- "good": +1.9
- "terrible": -2.1
- "!" : +0.2 intensity
- "but":  $\times 1.5$  weight shift
- Other words: 0.0

### Calculation

$$\begin{aligned} \text{Score} &= 1.9 + 1.5 \times (-2.1 + 0.2) \\ &= 1.9 + 1.5 \times (-1.9) \\ &= 1.9 - 2.85 = -0.95 \\ \text{Normalized} &\rightarrow -0.35 \end{aligned}$$



© 2025 Bidur Devkota,PhD. CC BY-NC.

## Machine Learning Sentiment Analysis

### Core Idea

Learn patterns from labeled examples

### Training Phase

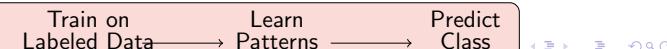
- Dataset: 50,000 labeled reviews
- Labels: Positive/Negative
- Goal: Learn statistical patterns

### Pattern Learning

- "good" → often positive
- "terrible" → strongly negative
- "X but Y terrible" → negative
- Learns from examples

### Prediction

- Doesn't understand words
- Recognizes patterns
- Uses statistical probabilities
- Output: Negative ( $P=0.92$ )



© 2025 Bidur Devkota,PhD. CC BY-NC.

## Lexicon Approach: Key Points

### How It Works

- Fixed dictionary of words
- Each word has sentiment score
- Apply linguistic rules
- Sum scores with weights

### Cons

- Static dictionary
- Limited context
- No learning
- Dictionary bias

### Pros

- Fast - no training
- Transparent
- Consistent
- Lightweight

### Use Cases

- Quick analysis
- Small datasets
- Prototyping
- Multi-language (with dictionaries)

### Thought Process:

"I found 'good' (+1.9) and 'terrible' (-2.1) in my dictionary. After rules, negative wins."

© 2025 Bidur Devkota,PhD. CC BY-NC.

## ML Approach: Key Points

### How It Works

- Feature extraction
- Train on labeled data
- Learn statistical patterns
- Probability-based prediction

### Cons

- Needs labeled data
- Black box
- Computationally heavy
- Training bias

### Pros

- Adaptive - learns
- Context aware
- Handles nuances
- Improves with data

### Use Cases

- Large datasets
- Complex patterns
- Context matters
- Continuous improvement

### Thought Process:

"I've seen 5,000 'X but Y terrible' patterns. 95% were negative, so this is negative too."

© 2025 Bidur Devkota,PhD. CC BY-NC.

## Comparison: Lexicon vs Machine Learning

Lexicon-Based		vs	Machine Learning	
Workflow		Workflow		
1. Dictionary lookup 2. Apply rules 3. Calculate score		1. Train on data 2. Learn patterns 3. Predict class		
Pros	Cons	Pros	Cons	
Fast Transparent No training Lightweight	Static Limited context Dictionary bias No learning	Adaptive Context aware Accurate Learns nuances	Data hungry Black box Computationally heavy Training bias	

### Modern Approach: Hybrid Systems

Use lexicon for speed + ML for complex cases. Lexicon scores as features for ML models.

© 2025 Bidur Devkota, PhD. CC BY-NC.

## Mathematical Comparison

### Lexicon Formula

$$S = \sum_{i=1}^n w_i \cdot s_i + I$$

- $s_i$ : word sentiment score
- $w_i$ : weight (e.g., 1.5 after "but")
- $I$ : intensity modifiers

### ML Formula (Logistic Regression)

$$P = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

- $\mathbf{x}$ : feature vector
- $\mathbf{w}$ : learned weights
- $b$ : bias term
- $P$ : probability

### Our Example

$$\begin{aligned} S &= 1.9 + 1.5 \times (-2.1 + 0.2) \\ &= 1.9 - 2.85 = -0.95 \\ &\rightarrow -0.35 \end{aligned}$$

$$\mathbf{w}^T \mathbf{x} + b = -2.3$$

$$\begin{aligned} P &= \frac{1}{1 + e^{-2.3}} = 0.92 \\ &\rightarrow \text{Negative} \end{aligned}$$

© 2025 Bidur Devkota, PhD. CC BY-NC.