1. Why Preprocessing is Essential for Textual Data

The Problem with Raw Text

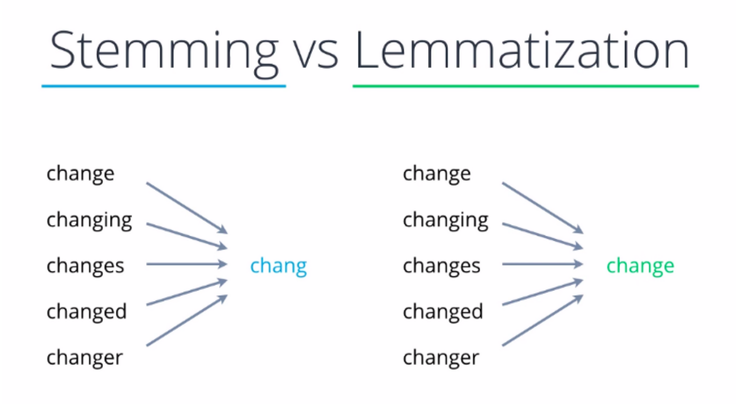
Raw text data contains numerous elements that don't contribute to meaningful analysis:

- **Noise**: Punctuation, special characters, numbers
- **Inconsistency**: Different cases (uppercase/lowercase)
- **Common words**: Articles, prepositions that appear everywhere
- **Variations**: Same word in different forms (run, running, ran)

Benefits of Preprocessing

1. **Standardization**: Ensures consistent format across all text
2. **Noise Reduction**: Removes irrelevant characters and formatting
3. **Feature Quality**: Improves the quality of extracted features
4. **Computational Efficiency**: Reduces vocabulary size and processing time

2. Stemming vs. Lemmatization



Stemming

- **Definition**: Reduces words to their root form by removing suffixes
- **Method**: Uses rule-based algorithms (like Porter Stemmer)
- **Example**: "running" → "run", "better" → "better"
- **Pros**: Fast, simple
- **Cons**: May create non-words ("studies" → "studi")

Lemmatization

- **Definition**: Reduces words to their dictionary base form
- **Method**: Uses linguistic knowledge and dictionary

- **Example**: "running" → "run", "better" → "good"
- **Pros**: Produces real words, more accurate
- **Cons**: Slower, requires more resources

3. Why Feature Extraction is Necessary

The Challenge

Computers cannot directly process text - they need numerical representations. Raw text has:

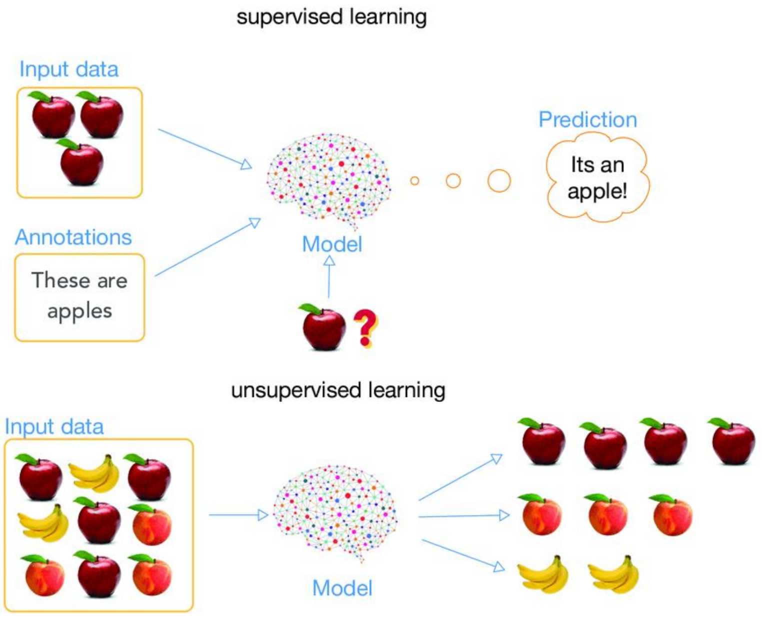
- **High dimensionality**: Each unique word could be a feature
- **Sparsity**: Most words don't appear in most documents
- **No semantic understanding**: "good" and "excellent" are treated as completely different

The Solution: Feature Vectors

Feature extraction converts text into numerical vectors that:

- **Capture semantic meaning**: Similar texts have similar vectors
- **Enable mathematical operations**: Clustering, classification, similarity calculation
- **Reduce dimensionality**: From thousands of words to hundreds of features

4. Supervised vs. Unsupervised Learning



Supervised Learning

- **Definition**: Learning with labeled training data

- **Goal**: Predict labels for new data
- **Examples**: Classification, regression
- **Evaluation**: Compare predictions to true labels
- **Requirement**: Need labeled dataset

Unsupervised Learning

- **Definition**: Learning without labeled data
- **Goal**: Discover hidden patterns in data
- **Examples**: Clustering, dimensionality reduction
- **Evaluation**: Internal metrics (silhouette score)
- **Advantage**: No need for labeled data

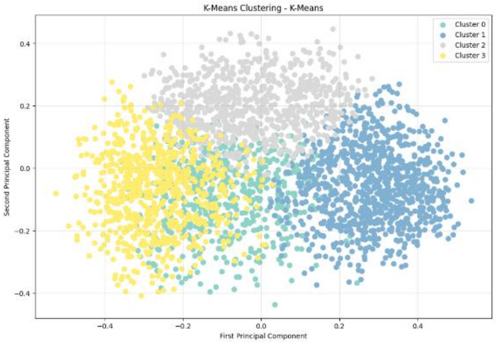
Comparison

Aspect	Supervised	Unsupervised
Labels	Required	Not required
Goal	Prediction	Pattern discovery
Evaluation	External metrics	Internal metrics
Complexity	Generally easier	More challenging

5. Clustering Algorithms Explained

K-Means Clustering

- **Principle**: Partitions data into K clusters by minimizing within-cluster variance
- **Algorithm**:
  1. Choose K cluster centers randomly
  2. Assign each point to nearest center
  3. Update centers to cluster centroids
  4. Repeat until convergence
- **Advantages**: Simple, fast, works well with spherical clusters
- **Disadvantages**: Need to specify K, assumes spherical clusters

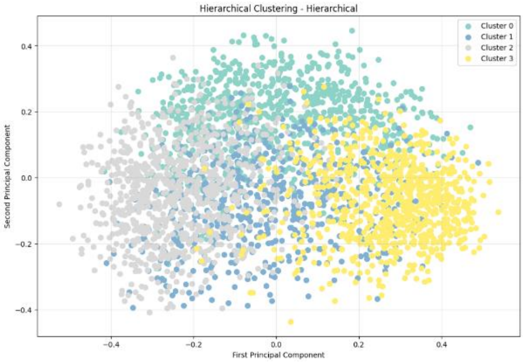


(Density-Based Spatial Clustering)

- **Principle**: Groups points that are closely packed, marks outliers
- **Parameters**:
  - eps: Maximum distance between two samples
  - min_samples: Minimum number of samples in neighborhood
- **Advantages**: Finds clusters of arbitrary shape, identifies outliers
- **Disadvantages**: Sensitive to parameters, struggles with varying densities

Hierarchical Clustering

- **Principle**: Creates tree of clusters by merging or splitting
- **Types**:
  - **Agglomerative**: Bottom-up (merge clusters)
  - **Divisive**: Top-down (split clusters)
- **Advantages**: No need to specify number of clusters, creates hierarchy
- **Disadvantages**: Computationally expensive, sensitive to noise



6. Text Vectorization Methods

Bag of Words (BoW)

- **Method**: Count frequency of each word
- **Pros**: Simple, interpretable
- **Cons**: Ignores word order, creates sparse vectors

TF-IDF (Term Frequency-Inverse Document Frequency)

- **Method**: Weighs terms by frequency and rarity
- **Formula**: TF-IDF = TF × log(N/DF)
- **Pros**: Reduces impact of common words
- **Cons**: Still sparse, no semantic understanding

Word Embeddings (Word2Vec, GloVe)

- **Method**: Dense vectors that capture semantic relationships
- **Pros**: Captures word relationships, dense vectors
- **Cons**: Fixed vocabulary, single vector per word

Sentence Transformers

- **Method**: Neural networks that create sentence-level embeddings
- **Pros**: Captures full sentence meaning, contextual understanding
- **Cons**: Computationally intensive, requires pre-training

7. SentenceTransformers and all-MiniLM-L6-v2

SentenceTransformers Framework

- **Purpose**: Create dense vector representations of sentences
- **Architecture**: Based on transformer models (BERT, RoBERTa)
- **Training**: Siamese networks with semantic similarity tasks
- **Output**: Fixed-size vectors (typically 384 or 768 dimensions)

all-MiniLM-L6-v2 Model

- **Type**: Lightweight transformer model
- **Architecture**: 6-layer MiniLM (distilled BERT)
- **Output Size**: 384 dimensions
- **Performance**: Good balance of speed and quality
- **Use Case**: Ideal for semantic similarity and clustering tasks

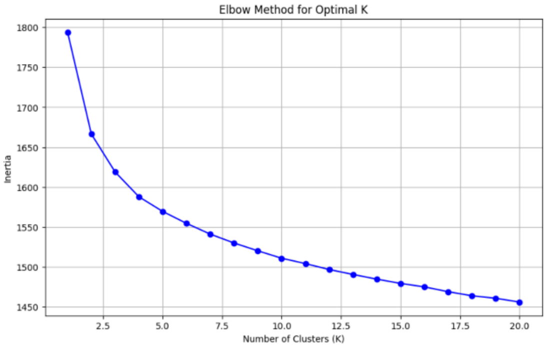
8. Elbow Method for K-Means

Purpose

Determine optimal number of clusters (K) for K-Means algorithm

Method

1. Run K-Means for different values of K (1 to max_k)
2. Calculate inertia (sum of squared distances from points to centroids)
3. Plot K vs. inertia
4. Look for "elbow" point where inertia reduction slows



Interpretation

- **Sharp elbow**: Clear optimal K
- **Gradual curve**: Less clear, use domain knowledge
- **Multiple elbows**: Consider multiple valid K values

9. Principal Component Analysis (PCA)

Purpose

Reduce dimensionality while preserving most variance in data

How It Works

1. **Center the data**: Subtract mean from each feature
2. **Calculate covariance matrix**: Measure feature relationships
3. **Find eigenvectors**: Directions of maximum variance
4. **Project data**: Transform to new coordinate system

Benefits

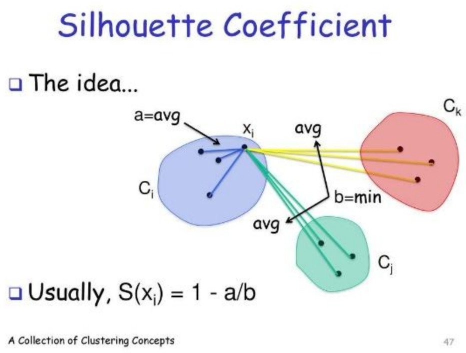
- **Dimensionality reduction**: From hundreds to 2-3 dimensions
- **Visualization**: Can plot high-dimensional data
- **Noise reduction**: Focuses on main patterns

Limitations

- **Interpretability**: New dimensions are combinations of original features
- **Linear**: Cannot capture non-linear relationships
- **Information loss**: Some variance is always lost

10. Evaluation Metrics

Silhouette Score



Silhouette Coefficient

❑ The idea...

❑ Usually, $S(x_i) = 1 - a/b$

A Collection of Clustering Concepts

47

- **Range**: -1 to 1
- **Interpretation**:
  - Close to 1: Well-separated clusters
  - Close to 0: Overlapping clusters
  - Close to -1: Incorrectly clustered
- **Calculation**: For each point, compare distance to own cluster vs. nearest cluster

Homogeneity Score

$$h = 1 - \frac{H(Y_{true}|Y_{pred})}{H(Y_{true})}$$

- **Range**: 0 to 1
- **Interpretation**: How pure are the clusters (if true labels known)
- **Calculation**: Based on conditional entropy
- **Note**: Requires ground truth labels (not available for our unsupervised task)

Why Homogeneity Can't Be Used

In our project, we don't have ground truth labels for song genres or themes. Homogeneity requires knowing the true cluster assignments, making it unsuitable for purely unsupervised clustering tasks.

KMeans Cluster Analysis and Interpretation

Based on the sample lyrics from each cluster, here's what your clustering algorithm has discovered:

Cluster 0 (Size: 584) - Personal/Emotional Themes

Characteristics:

- **Themes**: Personal struggles, relationships, inner thoughts
- **Language**: Conversational, introspective, emotional
- **Topics**: Marriage, therapy, personal growth, clarity, contemplation
- **Style**: Modern, relatable, everyday language

Sample Analysis:

- Sample 1: Relationship and personal struggles ("need therapy", "wan na marry")
- Sample 2: Change and contemplation ("time changing", "complicated contemplating")
- Sample 3: Inner growth and persistence ("place inside soul", "getting bigger")
- Sample 4: War/conflict themes (could be metaphorical for personal battles)

Cluster 1 (Size: 934) - Hip-Hop/Rap Culture

Characteristics:

- **Themes**: Party culture, street life, bragging rights, urban lifestyle
- **Language**: Slang, rap terminology, aggressive/confident tone
- **Topics**: Parties, MC culture, street credibility, rap battles
- **Style**: Rhythmic, urban vernacular, hip-hop specific references

Sample Analysis:

- Sample 1: Party culture ("party people", "MC icet", "syndicate")
- Sample 2: Street credibility and rap culture ("overdose overload", "rap visible")
- Sample 3: Confrontational/bragging ("acting brand new", "drunk dialin")
- Sample 4: Urban references ("magic johnson", "compton", street terminology)

Cluster 2 (Size: 757) - Love/Romance/Pop Themes

Characteristics:

- **Themes**: Love, relationships, longing, emotional connection
- **Language**: Romantic, tender, emotional, accessible
- **Topics**: Home, family, love, dreams, waiting for someone
- **Style**: Pop-oriented, mainstream appeal, emotional storytelling

Sample Analysis:

- Sample 1: Family and belonging ("home place roam", "mother sister brother")
- Sample 2: Romantic imagery ("wrap sky", "carve name", "beautiful impossible gift")
- Sample 3: Emotional pop themes ("used happy", "feel low", collaboration style)
- Sample 4: Classic love song themes ("waiting life", "dreamed holding tight")

Cluster 3 (Size: 724) - Dark/Gothic/Metal Themes

Characteristics:

- **Themes**: Darkness, death, existential dread, supernatural elements
- **Language**: Complex, poetic, metaphorical, intense
- **Topics**: Death, evil, spiritual darkness, societal criticism
- **Style**: Gothic, metal, alternative, philosophical

Sample Analysis:

- Sample 1: Supernatural/dark imagery ("shadow darkest realm", "spirit oblivion")
- Sample 2: Abstract/surreal themes ("blisswet air", "colourize pain")
- Sample 3: Mystical/spiritual elements ("tunnel worm dove", "thirteen moon temple")
- Sample 4: Social criticism and darkness ("hatred", "society blame", "bloodshed")

Clustering Success Analysis

Why This Clustering Works Well:

1. **Clear Thematic Separation**: Each cluster represents a distinct musical genre/theme
2. **Semantic Coherence**: Songs within each cluster share similar vocabulary and themes
3. **Balanced Distribution**: Clusters have reasonable sizes (584, 934, 757, 724)
4. **Genre Recognition**: The algorithm successfully identified major music genres

Genre Mapping:

- **Cluster 0**: Pop/Contemporary (Personal themes)
- **Cluster 1**: Hip-Hop/Rap (Urban culture)
- **Cluster 2**: Pop/Love Songs (Romantic themes)
- **Cluster 3**: Metal/Gothic/Alternative (Dark themes)

Musical Genre Characteristics Captured

Lyrical Patterns Identified:

Cluster	Genre	Key Indicators	Vocabulary Style
0	Contemporary Pop	Personal pronouns, emotions	Conversational
1	Hip-Hop/Rap	Street slang, party references	Urban vernacular
2	Love/Romance	Relationship terms, tender language	Emotional, accessible
3	Metal/Gothic	Dark imagery, complex metaphors	Poetic, intense

Semantic Similarity Success:

- **Within-cluster similarity**: High (songs share thematic content)
- **Between-cluster distinction**: Clear (different vocabularies and themes)
- **Cultural relevance**: Clusters align with real music genre

---

Hierarchical Cluster Analysis and Interpretation

Cluster 0 (803 songs): Personal transformation and introspective themes

- Keywords: "change life," "exorcise demon," "memory," "dream"

Cluster 1 (582 songs): Social commentary and rebellion

- Keywords: "imagination," "disease," "technology," "fight," "hatred"

Cluster 2 (837 songs): Narrative and storytelling

- Keywords: "facade," "pillow dream," "strong horse," "beyond"

Cluster 3 (777 songs): Hip-hop/rap style with lifestyle themes

- Keywords: "showbiz," "late night," "bill," "work," "money"