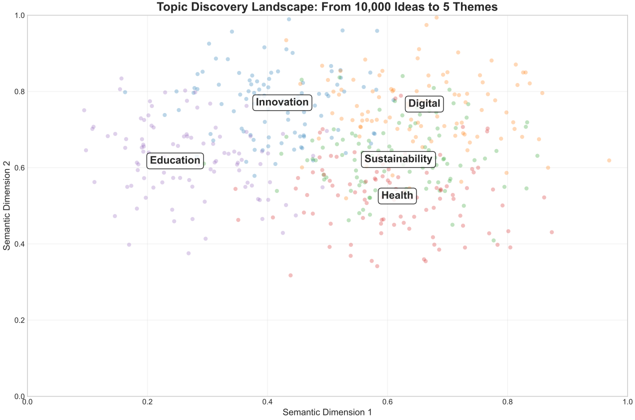
Week 5: Topic Modeling for Ideation

Design & AI Program

Agenda



10,000 Ideas

How do we find patterns?

- Customer feedback: 5,000 reviews
- Innovation workshops: 2,000 ideas
- Market research: 3,000 insights

Topic modeling reveals hidden structure

The Ideation Problem

Traditional Ideation

- Manual categorization
- Subjective grouping
- Limited scale (100s of ideas)
- Bias toward obvious themes
- Missing connections

Result: Lost opportunities

ML-Enhanced Ideation

- Automatic theme discovery
- Data-driven clustering
- Unlimited scale (1000s+)
- Hidden pattern detection
- Cross-theme insights

Result: Innovation patterns

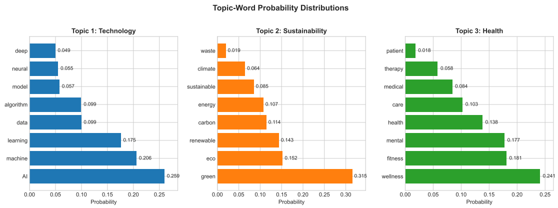
Topic modeling bridges human creativity and machine pattern recognition

Core Concept

Finding hidden themes in large text collections

Key Assumptions:

1. Documents contain multiple topics
2. Topics are probability distributions over words
3. Words can belong to multiple topics



Each document is a mixture of topics, each topic is a mixture of words

Real-World Impact

Netflix
Content categorization
35 micro-genres discovered
+18% engagement

IDEO
Design research synthesis
60% faster insights
3x more patterns

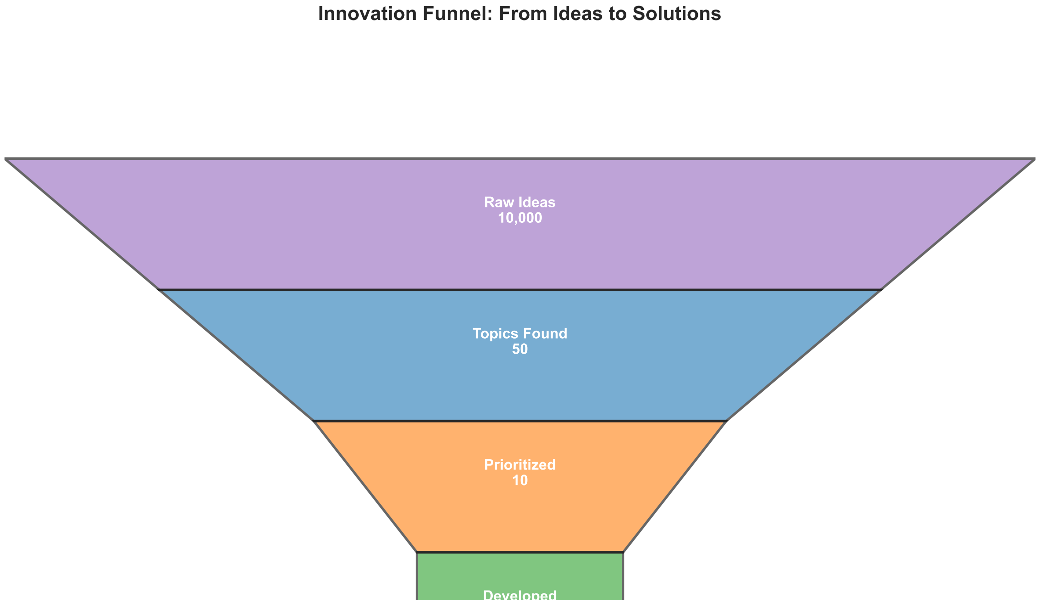
Spotify
Music recommendation
1,500 micro-moods
+25% listening time

3M
Innovation mining
47 new product ideas
$12M revenue

P&G
Consumer needs analysis
23 unmet needs found
5 new products

Amazon
Review insights
Product improvements
-30% returns

Topic modeling transforms unstructured data into actionable innovation insights


Innovation Funnel: From Ideas to Solutions
Raw Ideas
10,000
Topics Found
50
Prioritized
10
Developed

Topic Modeling Approaches

Probabilistic Models

LDA (Latent Dirichlet Allocation)

- Bayesian approach
- Interpretable topics
- Industry standard

LSA (Latent Semantic Analysis)

- Matrix factorization
- Fast computation
- Good for similarity

Modern Approaches

NMF (Non-negative Matrix Factorization)

- Parts-based representation
- Sparse, interpretable
- Good for short texts

BERTopic

- Transformer-based
- Contextual understanding
- State-of-the-art accuracy

Choose based on: data size, interpretability needs, computational resources

Technical Skills

1. Build topic models with LDA/NMF
2. Optimize hyperparameters
3. Visualize topic distributions
4. Evaluate model quality
5. Handle different text types

Design Applications

1. Transform ideas into themes
2. Identify innovation patterns
3. Create opportunity maps
4. Prioritize based on topics
5. Generate new combinations

Outcome: Data-driven ideation at scale

Foundation Summary

Key Concepts

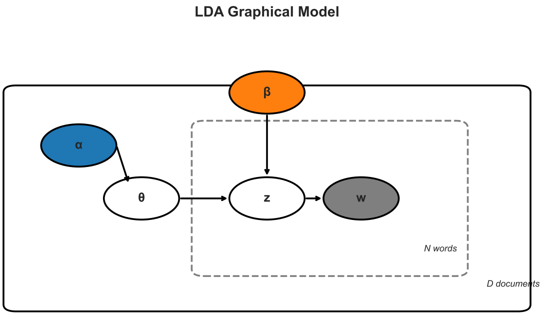
- Topics as hidden themes
- Probabilistic word distributions
- Document-topic mixtures
- Unsupervised discovery

Business Value

- Scale ideation 100x
- Find non-obvious patterns
- Reduce bias in categorization
- Accelerate innovation cycles

Next: Deep dive into algorithms

Latent Dirichlet Allocation (LDA)



Generative Process:

1. Choose topic distribution for document
2. For each word position:
   - Choose a topic
   - Choose a word from that topic

Key Parameters

- **K**: Number of topics
- α: Document-topic density
- β: Topic-word density

Strengths:

- Probabilistic interpretation
- Handles uncertainty
- Industry standard

Limitations:

- Fixed number of topics
- Assumes bag-of-words

LDA Intuition: Restaurant Reviews Example

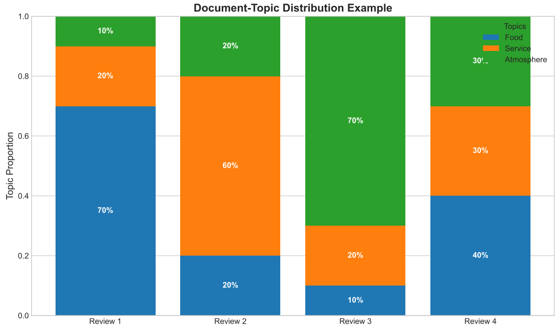
Input Reviews:
"The pasta was delicious and service excellent"
"Great atmosphere, loved the wine selection"
"Fast delivery, food arrived hot"
"Cozy ambiance, romantic lighting"

LDA Discovers:

Topic 1: Food Quality
pasta, delicious, hot, fresh, taste

Topic 2: Service
service, delivery, fast, staff, friendly

Topic 3: Atmosphere
atmosphere, ambiance, cozy, romantic



Each review contains multiple topics in different proportions

Non-negative Matrix Factorization (NMF)

Core Concept

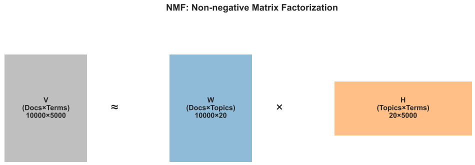
Decompose document-term matrix:

$\mathbf{V} = \mathbf{W} \times \mathbf{H}$

- **V**: Original documents
- **W**: Document-topic weights
- **H**: Topic-term weights

Key Difference from LDA:

- Deterministic
- Parts-based representation
- No probabilistic assumptions



When to Use:

- Short texts (tweets, titles)
- Need interpretable parts
- Speed is critical
- Sparse data

Latent Semantic Analysis (LSA)

Singular Value Decomposition

$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$

- Reduces dimensionality
- Captures semantic relationships
- Handles synonyms naturally

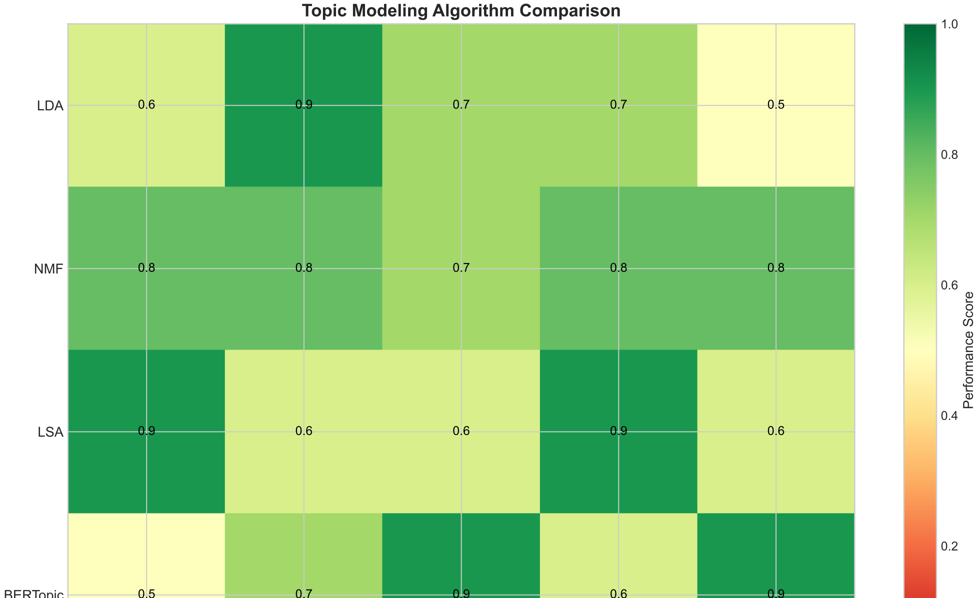
Process:

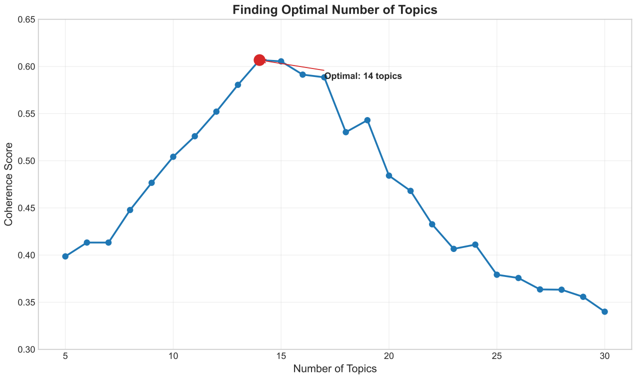
1. Build term-document matrix
2. Apply TF-IDF weighting
3. Perform SVD
4. Keep top k dimensions

lsa_semantic_space.pdf

Applications:

Algorithm Comparison





Practical Guidelines

Rule of Thumb:

$K = \sqrt{N/2}$ for N documents

Domain-Specific:

- News: 20-50 topics
- Reviews: 5-15 topics
- Research: 30-100 topics
- Social media: 10-30 topics

Best Practice:

Test multiple values, validate with domain experts

Metrics to Consider:

- **Coherence**: Semantic similarity
- **Perplexity**: Model fit
- **Distinctiveness**: Topic separation

Coherence Measures

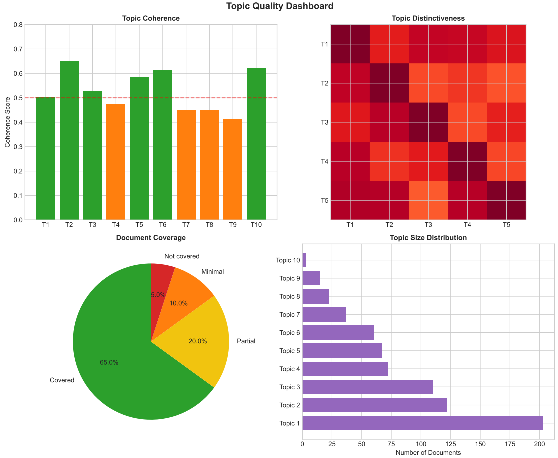
UMass Coherence:
Based on document co-occurrence

C_V Coherence:
Sliding window + word embeddings

C_NPMI:
Normalized pointwise mutual information

Human Evaluation

- Topic interpretability
- Word intrusion test
- Topic intrusion test



Quality Thresholds:

- Coherence > 0.5: Good
- Distinctiveness > 0.7: Good
- Coverage $> 80\%$: Good

Modern Approach: BERTopic

Pipeline:

1. Embed documents (BERT)
2. Cluster embeddings (HDBSCAN)
3. Create topics (c-TF-IDF)
4. Fine-tune representations

Advantages:

- Contextual understanding
- Dynamic number of topics
- Outlier detection
- Hierarchical topics

bertopic_pipeline.pdf

Use Cases:

Preprocessing for Topic Models

Essential Steps:

1. **Tokenization**
   Split into meaningful units
2. **Lowercasing**
   Normalize text
3. **Remove stopwords**
   Filter common words
4. **Lemmatization**
   Reduce to base forms
5. **Filter extremes**
   Remove rare/common terms

Advanced Techniques:

- **Bigrams/Trigrams**
  "machine learning" as one token
- **Named Entity Recognition**
  Preserve "New York"
- **Part-of-speech filtering**
  Keep nouns and verbs
- **Domain stopwords**
  Remove domain-specific noise

Quality preprocessing = Better topics

Algorithm Summary

Classical Methods

- **LDA**: Probabilistic gold standard
- **NMF**: Fast and interpretable
- **LSA**: Semantic relationships

Modern Methods

- **BERTopic**: Contextual understanding
- **Top2Vec**: Document embeddings
- **CTM**: Correlated topics

Selection Criteria

- Data size and type
- Interpretability needs
- Computational resources
- Real-time requirements
- Language complexity

Remember:
No single best algorithm - choose based on your specific use case

Next: Implementation in practice

Data Collection for Ideation

Internal Sources

- Innovation workshops notes
- Employee suggestions
- R&D documentation
- Meeting transcripts
- Project proposals

External Sources

- Customer feedback
- Social media mentions
- Competitor analysis
- Patent databases
- Academic research

data_sources_hierarchy.pdf

Volume Guidelines:

LDA Implementation with Gensim

```
from gensim import corpora, models
import pandas as pd
# Load and preprocess
docs = load_innovation_data()
texts = [preprocess(doc) for doc in docs]
# Create dictionary and corpus
dictionary = corpora.Dictionary(texts)
corpus = [dictionary.doc2bow(text)
          for text in texts]
# Build LDA model
lda_model = models.LdaModel(
    corpus=corpus,
    id2word=dictionary,
    num_topics=20,
    alpha='auto',
    eta='auto',
    passes=10,
    random_state=42
)
# Get topics
topics = lda_model.print_topics(
    num_words=10
)
```

Key Parameters

- `num_topics`: Start with $\sqrt{N/2}$
- `alpha`: Document-topic density
- `eta`: Topic-word density
- `passes`: Training iterations

Output Example: `Topic 0: sustainability, eco, green, renewable...`
`Topic 1: digital, app, mobile, user...`
`Topic 2: health, wellness, fitness...`

NMF Implementation with Scikit-learn

```
from sklearn.decomposition import NMF
from sklearn.feature_extraction.text
    import TfidfVectorizer
# Vectorize documents
vectorizer = TfidfVectorizer(
    max_features=1000,
    min_df=5,
    max_df=0.8,
    ngram_range=(1, 2)
)
doc_term_matrix = vectorizer.fit_transform(
    documents
)
# Apply NMF
nmf = NMF(
    n_components=15,
    init='nndsvd',
    max_iter=200,
    random_state=42
)
W = nmf.fit_transform(doc_term_matrix)
H = nmf.components_
# Extract topics
feature_names = vectorizer.get_feature_names_out()
top_words = extract_top_words(
    H, feature_names, n_top_words=10
)
```

Advantages for Ideation:

- Faster than LDA
- Better for short texts
- More stable results
- Easier interpretation

Best Practices:

- Use TF-IDF weighting
- Include bigrams
- Filter extremes carefully
- Validate with coherence

Visualizing Topics

pyLDAvis

pyldavis_example.pdf

Interactive topic exploration:

Custom Visualizations

topic_heatmap.pdf

Design-focused views:

Production Pipeline

production_pipeline.pdf

hyperparameter_grid.pdf

Tuning Strategy

1. **Number of topics**
   Try: 5, 10, 15, 20, 25, 30
2. **Alpha parameter**
   Try: auto, 0.01, 0.1, 1.0
3. **Beta/Eta parameter**
   Try: auto, 0.01, 0.1

Validation:

- Coherence score
- Human evaluation
- Business relevance

Adapting to Different Text Types

Short Text
(Tweets, titles)

- Use NMF or BERTopic
- Aggressive filtering
- Include hashtags
- Expand with context

Long Documents
(Reports, articles)

- LDA works well
- Paragraph sampling
- More topics needed
- Section-aware

Mixed Format
(Reviews + comments)

- Normalize lengths
- Weight by importance
- Hierarchical topics
- Multi-level models

Adapt preprocessing and model choice to your data characteristics

Real-time Topic Discovery

Online Learning

Update models incrementally:

- Online LDA
- Mini-batch processing
- Sliding window approach
- Dynamic topic models

Architecture:

- Message queues (Kafka)
- Stream processing (Spark)
- Model serving (MLflow)
- Result caching (Redis)

```
realtime_architecture.pdf
```

Use Cases:

Common Implementation Pitfalls

Mistakes to Avoid

1. **Too few documents**
   Need 100+ per expected topic
2. **No preprocessing**
   Garbage in, garbage out
3. **Wrong model choice**
   Match model to data type
4. **Fixed hyperparameters**
   Always tune and validate
5. **Ignoring domain knowledge**
   Involve subject experts

Best Practices

1. **Start simple**
   Basic LDA, then iterate
2. **Validate thoroughly**
   Multiple metrics + humans
3. **Document everything**
   Preprocessing, parameters
4. **Version control models**
   Track changes over time
5. **Monitor drift**
   Topics evolve, retrain

Success = Good data + Right model + Careful validation

Implementation Checklist

Data Preparation ✓

- Collect diverse sources
- Clean and preprocess
- Create document-term matrix
- Split train/validation

Model Development ✓

- Choose algorithm
- Tune hyperparameters
- Validate coherence
- Interpret topics

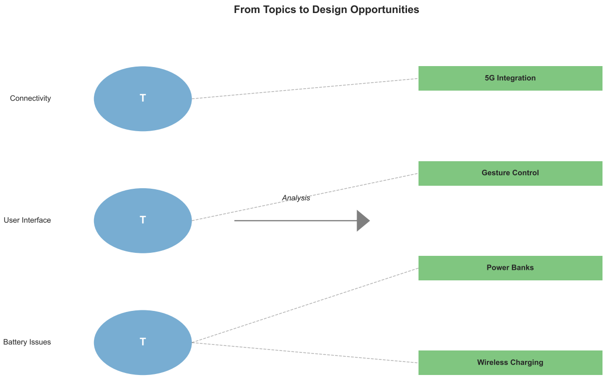
Deployment ✓

- Build pipeline
- Create visualizations
- Set up monitoring
- Document API

Maintenance ✓

- Track performance
- Update regularly
- Gather feedback
- Iterate and improve

Next: Applying topics to design



Example Translation

Topic: "battery, charging, power, drain"

User Need:

Longer battery life, faster charging

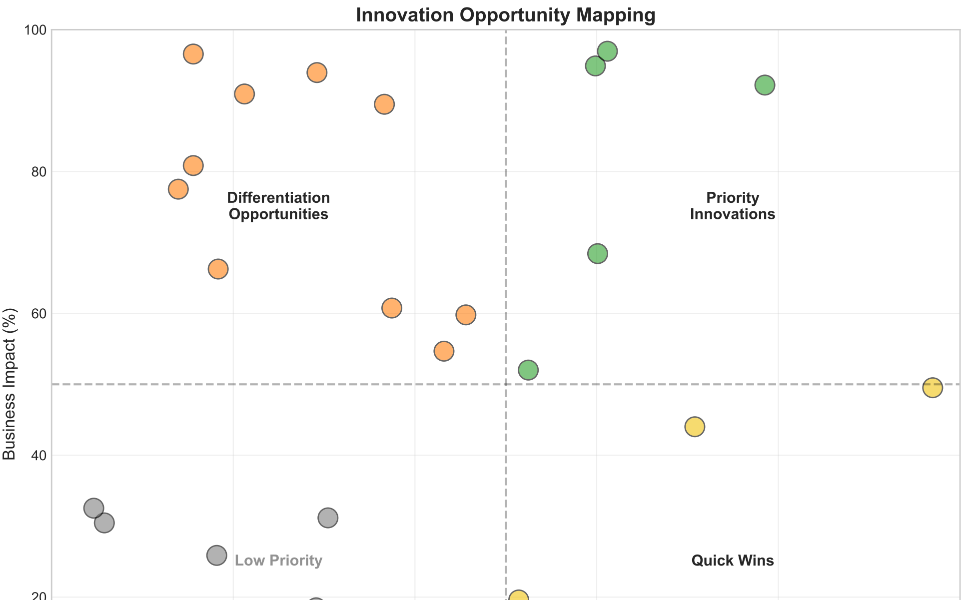
Design Opportunity:

- Wireless charging stations
- Power-saving modes
- Solar accessories
- Battery health monitoring

Each topic reveals multiple design directions

Translation Process:

1. Discover topic clusters
2. Identify user needs
3. Map to design spaces
4. Generate solutions


Innovation Opportunity Mapping
100
80
60
40
20
Business Impact (%)
Differentiation
Opportunities
Priority
Innovations
Low Priority
Quick Wins

Cross-Topic Innovation

Topic Intersections

Combine topics for breakthrough ideas:

Topic A: Sustainability
Topic B: Smart home
Innovation: Eco-smart home system

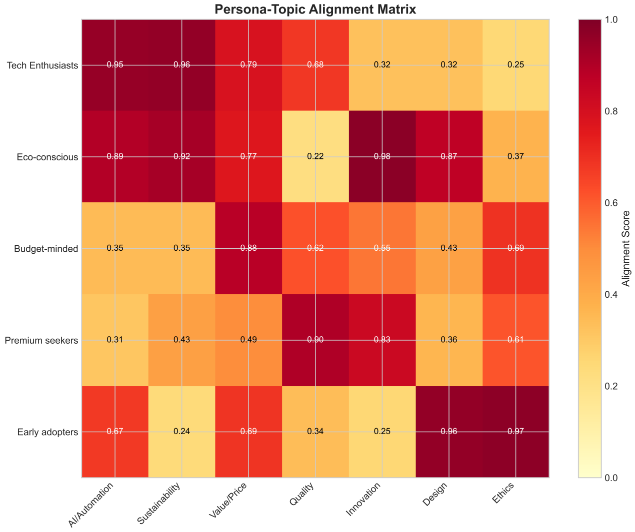
Topic C: Health tracking
Topic D: Gaming
Innovation: Fitness gamification

Topic E: Remote work
Topic F: Mental wellness
Innovation: Virtual wellness offices

topic_intersection_matrix.pdf

Innovation Formula:

Persona-Topic Alignment



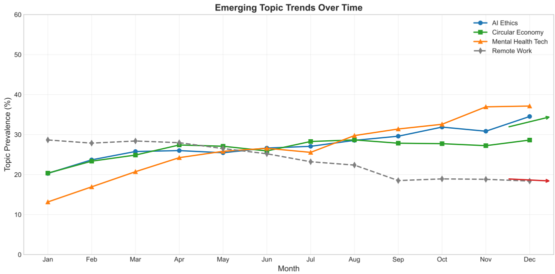
Design Strategy

1. Map topics to personas
2. Identify gaps and overlaps
3. Prioritize by segment size
4. Design targeted solutions

Benefits:

- Personalized innovation
- Better product-market fit
- Focused development
- Clear messaging

Insights:



Rising Topics:

- AI ethics (+45% monthly)
- Circular economy (+38%)
- Mental health tech (+52%)

Trend Analysis

Time-based Topic Modeling:

1. Weekly topic extraction
2. Track topic evolution
3. Identify emerging themes
4. Predict future directions

Action Items:

- Invest in rising topics
- Pivot from declining themes
- First-mover advantage
- Strategic positioning

Feature Prioritization Matrix

feature_priority_matrix.pdf

Priority Framework

Score = F × I × S

- F: Frequency (0-1)
- I: Impact (0-1)
- S: Sentiment (0-1)

Top Features:

1. Voice control (0.82)
2. Auto-save (0.78)
3. Dark mode (0.75)
4. Offline mode (0.71)
5. Collaboration (0.68)

Data-driven roadmap planning

Topic-Driven Innovation Workshops

Pre-Workshop

1. Run topic analysis
2. Identify top 10 themes
3. Create topic cards
4. Prepare inspiration boards

During Workshop

1. Present topic insights
2. Brainstorm per topic
3. Cross-pollinate themes
4. Vote and prioritize

Workshop Tools

workshop_toolkit.pdf

Competitive Intelligence Through Topics

competitive_topic_analysis.pdf

Analysis Process

1. Collect competitor data
2. Extract their topics
3. Compare topic distributions
4. Identify white spaces
5. Plan differentiation

Strategic Actions:

- Enter unserved topics
- Strengthen unique positions
- Avoid crowded spaces
- Create new categories

Gap Analysis:

ROI of Topic-Driven Innovation

Efficiency Gains

- 70% faster ideation
- 50% less redundancy
- 3x more ideas processed
- 60% cost reduction

Quality Improvements

- 40% better PMF
- 25% higher success rate
- 35% fewer pivots
- 2x user satisfaction

Business Impact

- 28% revenue growth
- 45% faster time-to-market
- 30% more patents filed
- 50% better retention

roi_timeline.pdf

Design Application Framework

Discovery Phase

- Extract topics from data
- Map to user needs
- Identify opportunities
- Analyze competition

Development Phase

- Prioritize features
- Design solutions
- Create prototypes
- Test with users

Delivery Phase

- Launch products
- Monitor feedback
- Track topic evolution
- Iterate and improve

Success Factors:

- Quality data sources
- Regular topic updates
- Cross-functional teams
- User validation

Next: Hands-on practice

Case Study: Smart Home Innovation

Challenge

A smart home company wants to identify new product opportunities from:

- 50,000 customer reviews
- 10,000 support tickets
- 5,000 forum discussions
- 2,000 survey responses

Goal: Find top 5 innovation opportunities

`case_study_overview.pdf`

Expected Outcomes:

- Topic map of user needs
- Priority feature list

Workshop Exercise: Your Turn

Dataset Provided

startup_ideas.csv

- 5,000 startup descriptions
- 15 industry categories
- Funding information
- Success metrics

Your Tasks:

1. Load and explore data
2. Preprocess text
3. Build topic model
4. Visualize results
5. Identify opportunities

Starter Code

```
import pandas as pd
from gensim import models
# Load data
df = pd.read_csv('startup_ideas.csv')
# Your code here:
# 1.  Preprocess descriptions
# 2.  Create topic model
# 3.  Extract insights
# 4.  Find patterns
# Deliverable:
# - 10 innovation themes
# - Top opportunities
# - Action plan
```

Time: 45 minutes

Step 1: Data Preparation `# Clean text`

```
def preprocess(text):
    # Lowercase
    # Remove special chars
    # Tokenize
    # Remove stopwords
    # Lemmatize
    return tokens
docs = df['description'].apply(preprocess)
```

Step 2: Build Model `# Create dictionary`

```
dictionary = corpora.Dictionary(docs)
corpus = [dictionary.doc2bow(doc)
          for doc in docs]
# Train LDA
lda = models.LdaModel(
    corpus, num_topics=15,
    id2word=dictionary
)
```

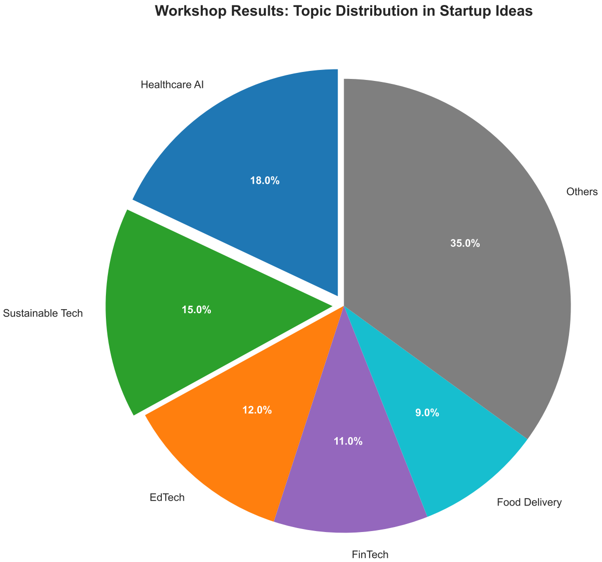
Step 3: Extract Insights `# Get topics`

```
for idx, topic in lda.print_topics():
    print(f'Topic {idx}: {topic}')
# Document-topic distribution
doc_topics = []
for doc in corpus:
    doc_topics.append(
        lda.get_document_topics(doc)
    )
```

Step 4: Visualize `# Create visualization`

```
import pyLDAvis.gensim
vis = pyLDAvis.gensim.prepare(
    lda, corpus, dictionary
)
pyLDAvis.display(vis)
```

Analyzing Your Results



Interpretation Guide

1. **Topic Prevalence**
   Which themes dominate?
2. **Topic Coherence**
   Do words make sense?
3. **Topic Distinctiveness**
   Are topics different?
4. **Business Relevance**
   Can we act on this?

Next Steps:

- Deep dive top 3 topics
- Cross-reference with trends
- Generate concepts

Innovation Opportunities Found

Opportunity 1
AI Health Assistant
Topics: Healthcare + AI + Elderly
Market size: $50B
Competition: Low
Feasibility: High

Priority: HIGH

Opportunity 2
Sustainable Packaging
Topics: Eco + Delivery + Waste
Market size: $30B
Competition: Medium
Feasibility: Medium

Priority: MEDIUM

Opportunity 3
EdTech Gamification
Topics: Education + Gaming + Kids
Market size: $20B
Competition: High
Feasibility: High

Priority: MEDIUM

opportunity_roadmap.pdf

Common Challenges & Solutions

Challenge 1: Messy Topics
Problem: Topics don't make sense
Solution:

- Better preprocessing
- Adjust number of topics
- Remove more stopwords
- Try different algorithms

Challenge 2: Overlapping Topics
Problem: Topics too similar
Solution:

- Reduce number of topics
- Increase alpha parameter
- Use hierarchical models

Challenge 3: Unactionable Insights
Problem: Topics not useful
Solution:

- Add domain knowledge
- Filter by relevance
- Combine with other data
- Involve stakeholders

Challenge 4: Scale Issues
Problem: Too slow on big data
Solution:

- Sample documents
- Use online learning
- Parallelize processing
- Cloud computing

Best Practices for Innovation Mining

Data Collection

- Diverse sources essential
- Quality over quantity
- Regular updates needed
- Include edge cases

Model Building

- Start simple, iterate
- Validate with humans
- Document parameters
- Version control models

Insight Generation

- Look for patterns
- Cross-reference topics
- Consider combinations
- Think user-first

Action Planning

- Prioritize ruthlessly
- Start with MVPs
- Test assumptions
- Measure impact

Success = Good Data + Right Model + Human Insight + Action

Key Takeaways

What You've Learned

1. Topic modeling transforms unstructured text into innovation insights
2. LDA, NMF, and modern methods each have strengths
3. Quality preprocessing is critical
4. Topics must translate to action
5. Combining topics creates breakthroughs

Your Toolkit

- ✓ Gensim for topic modeling
- ✓ pyLDAvis for visualization
- ✓ Evaluation metrics
- ✓ Design frameworks
- ✓ Workshop templates

You're ready to mine innovation at scale!

Next Week: Generative AI for Prototyping

week6_preview.pdf

Week 6 Preview

- GPT for design concepts
- DALL-E for visualization
- Code generation
- Rapid prototyping
- AI-assisted creativity

Preparation:

- Review transformer basics
- Explore GPT playground
- Think about prototypes

Generative Model

For each document d:

$$\theta_d \sim \mathrm{Dir}(\alpha)$$

For each word $w_{d,n}$ in document d:

$$z_{d,n} \sim \mathrm{Multinomial}(\theta_d)$$
$$w_{d,n} \sim \mathrm{Multinomial}(\beta_{z_{d,n}})$$

Where:

- θ_d: topic distribution for doc d
- $z_{d,n}$: topic for n-th word in doc d
- β_k: word distribution for topic k
- α: Dirichlet prior for documents

Posterior Inference

Goal: Estimate posterior

$$p(\theta, z | w, \alpha, \beta)$$

Approaches:

- Variational Inference (faster)
- Gibbs Sampling (more accurate)
- Online Learning (scalable)

Perplexity:

$$\text{Perplexity} = \exp\left(-\frac{\sum_d \log p(w_d)}{N}\right)$$

Lower is better

Matrix Factorization

$$\mathbf{V} \approx \mathbf{WH}$$

Where:

- $\mathbf{V} \in \mathbb{R}_+^{m \times n}$: document-term matrix
- $\mathbf{W} \in \mathbb{R}_+^{m \times k}$: document-topic matrix
- $\mathbf{H} \in \mathbb{R}_+^{k \times n}$: topic-term matrix
- All entries non-negative

Optimization:

$$\min_{W,H \geq 0} ||\mathbf{V} - \mathbf{WH}||_F^2$$

Update Rules

Multiplicative updates:

$$H_{ij} \leftarrow H_{ij} \frac{(W^T V)_{ij}}{(W^T WH)_{ij}}$$

$$W_{ij} \leftarrow W_{ij} \frac{(VH^T)_{ij}}{(WHH^T)_{ij}}$$

Convergence:

- Guaranteed to non-increase objective
- Local minimum (not global)
- Initialize multiple times

UMass Coherence

For top N words in topic:

$$C_{UMass} = \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{D(w_i, w_j) + \epsilon}{D(w_j)}$$

Where:

- $D(w_i, w_j)$: co-occurrence count
- $D(w_j)$: document frequency
- ϵ: smoothing parameter

C_V Coherence

Normalized PMI with sliding window

NPMI Coherence

$$NPMI(w_i, w_j) = \frac{\log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)}$$

Range: [-1, 1], higher is better

Topic Diversity

$$TD = \frac{|\text{unique words}|}{k \times N}$$

Where k = number of topics, N = words per topic

Kullback-Leibler Divergence

Between topic distributions:

$$D_{KL}(P||Q) = \sum_w P(w) \log \frac{P(w)}{Q(w)}$$

Used for:

- Topic distinctiveness
- Model comparison
- Variational inference

Mutual Information

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

Jensen-Shannon Divergence

Symmetric version of KL:

$$JSD(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M)$$

Where $M = \frac{1}{2}(P+Q)$

Topic Entropy

$$H(T) = -\sum_w P(w|T) \log P(w|T)$$

Lower entropy = more focused topic

Mathematical rigor ensures reliable innovation insights