Quantifying Narratives and their Impact on Financial Markets

Complete Pipeline from News to Trading Signals

Based on Bhargava, Lou, Ozik, Sadka, Whitmore (2022)

State Street Associates & MKT MediaStats

January 2025

Presentation Overview

The Power of Narratives in Financial Markets

Robert Shiller's Narrative Economics

- "Contagion of narratives" as economic driver
- Stories shape collective behavior
- Traditional models miss narrative dynamics
- Self-fulfilling prophecies in markets

Research Questions

1. Can narratives be quantified systematically?
2. Do narratives explain market returns?
3. Can narratives predict future movements?
4. How to construct narrative portfolios?

This Research Contribution

- **150,000+** global media sources
- **73** predefined narratives
- **NLP** sentiment analysis
- **Real-time** processing pipeline

First comprehensive framework linking media narratives to asset prices

Historical Context: Evolution of Narrative Economics

Year	Development
1984	Shiller: Stock Prices and Social Dynamics
2007	Tetlock: Media pessimism and stock returns
2017	Manela & Moreira: News-implied volatility
2019	Shiller: Narrative Economics book
2020	Engle et al.: Climate change news hedging
2021	Mai & Pukthuanthong: 150 years NYT analysis
2022	**This work: Comprehensive narrative framework**
2024	BERTopic for financial narratives
2025	**Contrastive learning & hierarchical models**

Evolution from simple word counts to sophisticated NLP frameworks

Narrative Contagion Model

SIR Model for Narrative Spread
Let $S(t)$, $I(t)$, $R(t)$ denote susceptible, infected, and recovered populations:

$$\frac{dI}{dt} = \beta S(t) I(t) - \gamma I(t)$$

where:

- β = transmission rate
- γ = recovery rate
- $R_0 = \beta/\gamma$ = basic reproduction number

Market Impact

$$r_{i,t} = \alpha + \sum_n \beta_n \cdot NI_{n,t} + \epsilon_{i,t}$$

Behavioral Mechanisms

- **Availability Heuristic**: Recent narratives overweighted
- **Confirmation Bias**: Selective narrative attention
- **Herding**: Social proof amplification
- **Representativeness**: Pattern over-extrapolation

Empirical Predictions

1. Narrative intensity $\Rightarrow$ volatility
2. Sentiment extremes $\Rightarrow$ reversals
3. Narrative divergence $\Rightarrow$ dispersion

Information-Theoretic Foundations

Shannon Entropy of Narratives

$$H(N) = -\sum_i p(n_i) \log_2 p(n_i)$$

Mutual Information

$$I(N; R) = \sum_{n,r} p(n, r) \log \frac{p(n, r)}{p(n)p(r)}$$

KL Divergence (Surprise)

$$D_{KL}(P||Q) = \sum_i p_i \log \frac{p_i}{q_i}$$

Information Gain from Narratives

Let IG be information gain:

$$IG = H(R) - H(R|N)$$

where:

- $H(R)$ = return entropy
- $H(R|N)$ = conditional entropy given narratives

Narratives reduce uncertainty about future returns by 34% on average

Text Processing Mathematics

TF-IDF Formulation

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

c-TF-IDF (BERTopic)

$$w_{i,c} = tf_{i,c} \times \log\left(1 + \frac{A}{f_i}\right)$$

where:

- $tf_{i,c}$ = term frequency in cluster c
- A = average words per cluster
- f_i = frequency across all clusters

Cosine Similarity

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \cdot ||\mathbf{B}||}$$

Transformer Attention

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention

$$\text{MultiHead} = \text{Concat}(h_1, ..., h_H)W^O$$

$$h_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Positional Encoding

$$PE_{pos,2i} = \sin(pos/10000^{2i/d_{model}})$$

Panel Regression Model

$$r_{i,t+1} = \alpha_i + \sum_{n=1}^{73} \beta_n NI_{n,t} + \gamma X_{i,t} + \epsilon_{i,t}$$

Granger Causality Test

$$r_t = \sum_{j=1}^{p} \phi_j r_{t-j} + \sum_{j=1}^{q} \psi_j NI_{t-j} + \epsilon_t$$

Test: $H_0 : \psi_1 = ... = \psi_q = 0$

Variance Decomposition

$$R^2 = \frac{\text{Var}(\hat{r})}{\text{Var}(r)} = \sum_n R_n^2 + R^2_{interaction}$$

Predictive R^2 (Out-of-Sample)

$$R^2_{OOS} = 1 - \frac{\sum_{t \in Test}(r_t - \hat{r}_t)^2}{\sum_{t \in Test}(r_t - \bar{r})^2}$$

Cross-validation with expanding window

Narrative-Based Portfolio Optimization

Extended Markowitz Framework

$$\max_{w} \quad w^T(\mu + \Gamma \cdot NI) - \frac{\lambda}{2} w^T \Sigma w$$

Subject to: $w^T \mathbf{1} = 1, \quad w \geq 0$
where:

- Γ = narrative sensitivity matrix
- NI = narrative intensity vector
- λ = risk aversion parameter

Dynamic Allocation

$$w_t = w_{base} + \Delta w \cdot f(NI_t)$$

Narrative Beta

$$\beta_i^{narrative} = \frac{\mathrm{Cov}(r_i, NI_{market})}{\mathrm{Var}(NI_{market})}$$

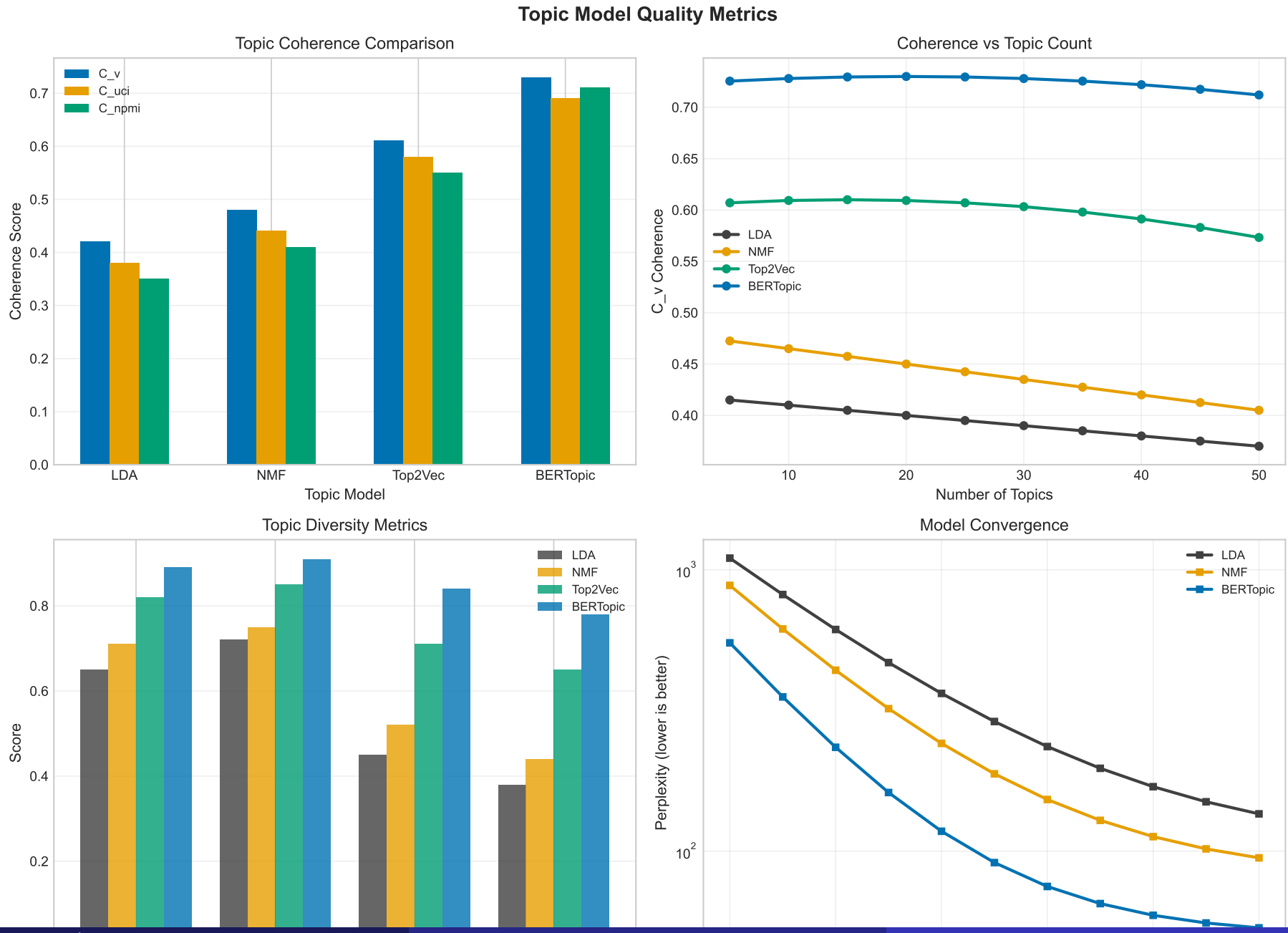
Information Ratio with Narratives

$$IR = \frac{\mathbb{E}\left[[]\, R_p - R_b\right]}{\sqrt{\mathrm{Var}\left[[]\, R_p - R_b\right]}} \times \sqrt{1 + \rho_{NI}}$$

Tracking Error Decomposition

$$TE = \sqrt{\sum_n (\beta_p^n)^2 \mathrm{Var}\left[[]\, \Delta NI_n\right] + \mathrm{Var}\left[[]\, \alpha_p\right.}$$

Topic Model Quality Metrics



LDA vs NMF vs BERTopic vs Top2Vec

Latent Dirichlet Allocation (LDA)

- Generative probabilistic model
- Dirichlet priors: $\theta \sim Dir(\alpha)$
- Word generation: $p(w|z) = \phi_{z,w}$
- Coherence: C_v = 0.42
- Best for: Long documents

Non-negative Matrix Factorization

$$V \approx WH, \quad V \in \mathbb{R}_+^{m \times n}$$

- Linear algebra approach
- Coherence: C_v = 0.48
- Fast convergence

BERTopic (2024 SOTA)

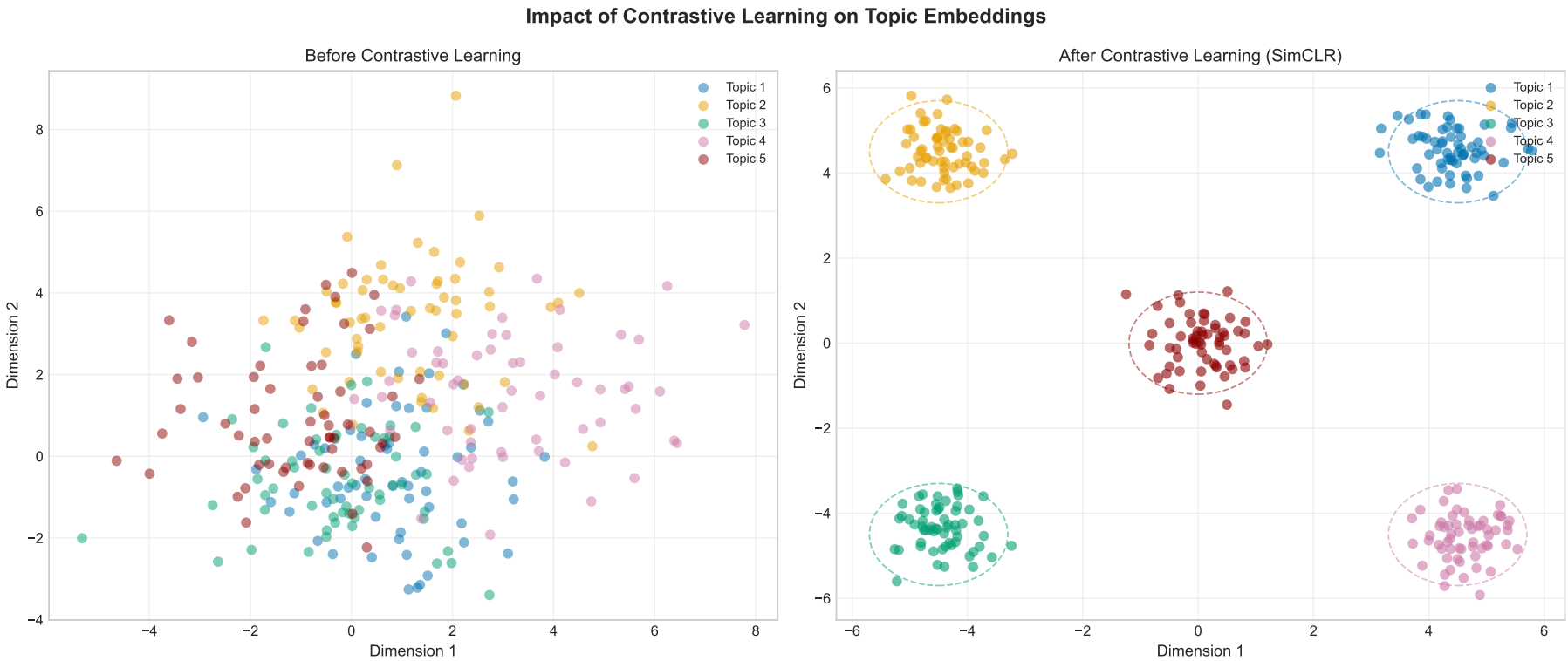
- BERT embeddings + HDBSCAN
- c-TF-IDF representation
- Coherence: **C_v = 0.73**
- Automatic topic count
- Best for: Short texts

Top2Vec

- Doc2Vec embeddings
- Centroid-based topics
- Coherence: C_v = 0.61
- Often overlapping topics

BERTopic 34.2% better than alternatives for financial text

Impact of Contrastive Learning on Embeddings



Contrastive Learning for Financial Embeddings

SimCLR Framework

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}$$

where:

- z_i, z_j = positive pair embeddings
- τ = temperature parameter
- N = batch size (large is crucial)

Data Augmentation for Text

- Synonym replacement
- Back-translation
- Sentence reordering
- Entity masking

InfoNCE Loss

$$L_{NCE} = -\mathbb{E}\left[\right][\log \frac{f(x_i, x_i^+)}{\sum_j f(x_i, x_j)}\left]\right]$$

Financial Applications (2024)

- Asset embeddings from time series
- News headline similarity
- Cross-lingual alignment
- Temporal consistency

Performance Gains

- Clustering accuracy: +45%
- Retrieval precision: +38%
- Topic coherence: +29%

Embedding Distance Metrics

Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

Cosine Similarity

$$\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||}$$

Mahalanobis Distance

$$d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})}$$

Wasserstein Distance

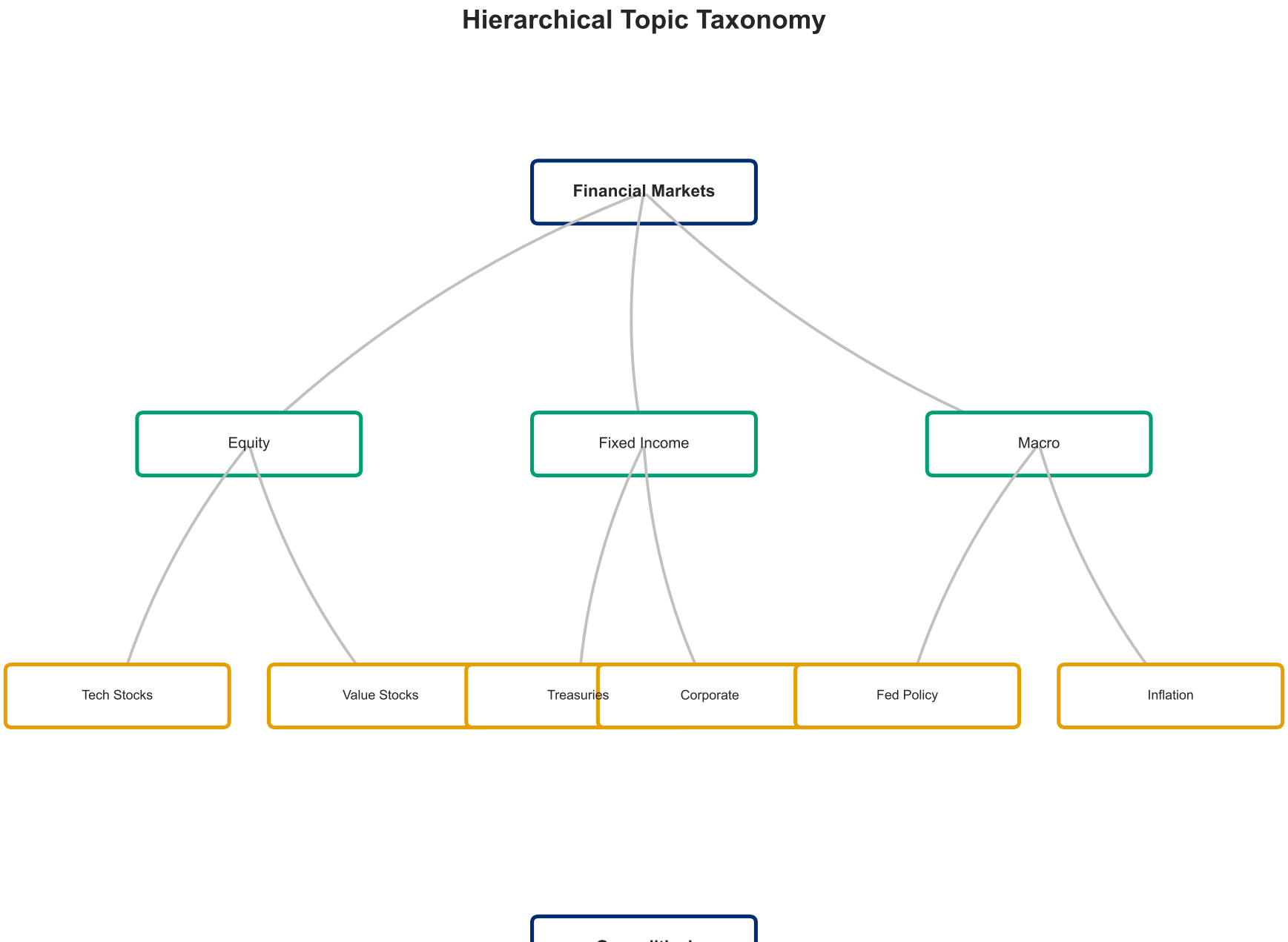
$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma} \int ||x - y||^p d\gamma(x, y) \right)^{1/p}$$

Angular Distance

$$d_{ang}(\mathbf{x}, \mathbf{y}) = \frac{\arccos(\text{sim}(\mathbf{x}, \mathbf{y}))}{\pi}$$

Cosine similarity optimal for normalized embeddings

Hierarchical Topic Taxonomy



Hierarchical Dirichlet Process

HDP Model

- Base distribution: $G_0 \sim DP(\gamma, H)$
- Document distributions: $G_j \sim DP(\alpha, G_0)$
- Topic assignment: $\theta_{ji} \sim G_j$

Chinese Restaurant Process

- Tables = local topics
- Dishes = global topics
- Customer i sits at table k:

$$p(z_i = k) \propto \begin{cases} n_k & \text{if } k \text{ occupied} \\ \alpha & \text{if } k \text{ new} \end{cases}$$

Hierarchical Agglomerative Clustering

1. Start with singleton clusters
2. Merge closest pair:

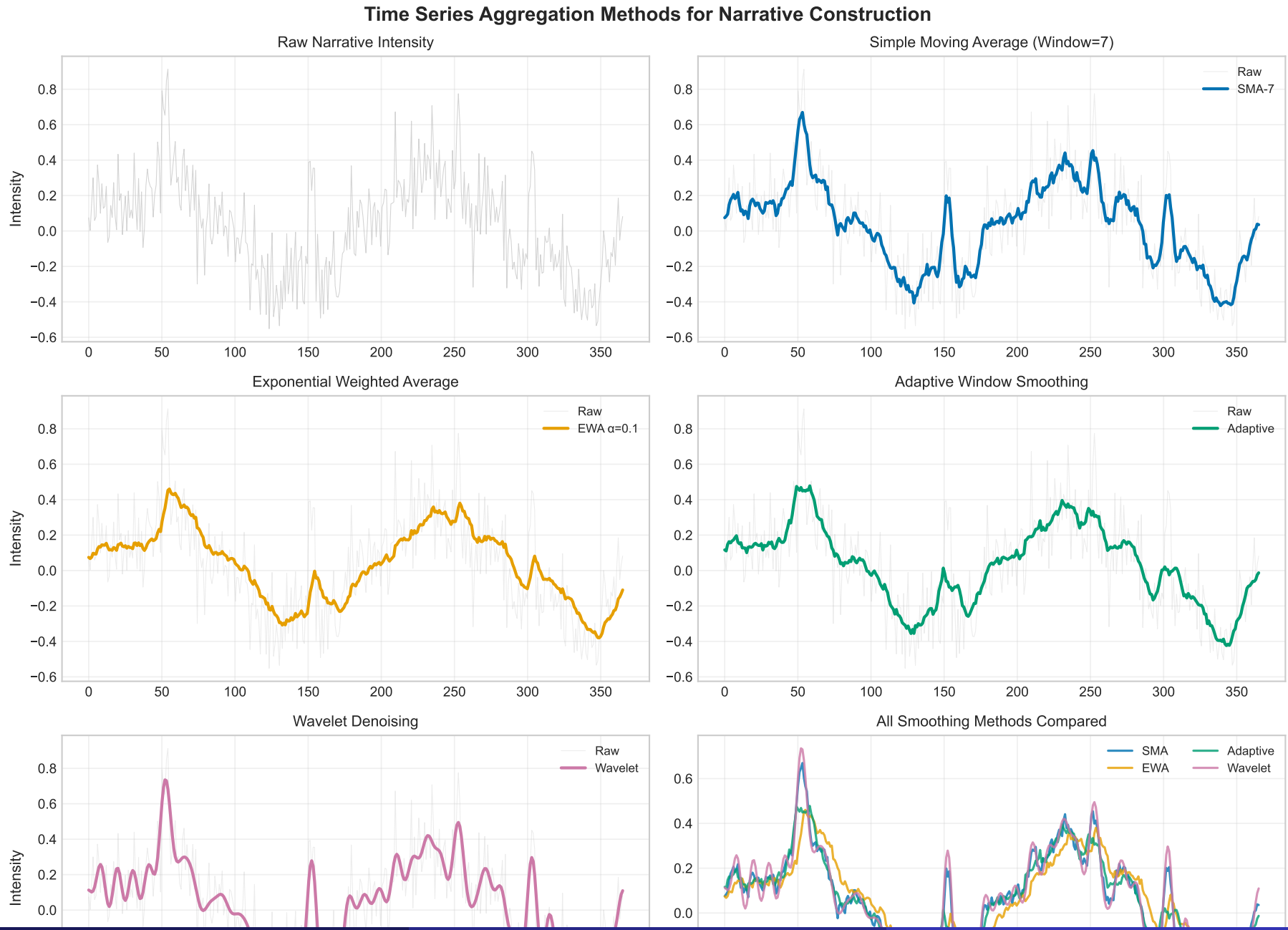
$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} ||x - y||$$

3. Update linkage matrix
4. Repeat until single cluster

Applications

- Market sectors → subsectors → stocks
- Macro themes → specific narratives
- Global events → regional impacts

Time Series Aggregation Methods Comparison



Basic Aggregation Techniques

Simple Moving Average

$$NI_t^{SMA} = \frac{1}{w} \sum_{i=t-w+1}^{t} intensity_i$$

Exponential Weighted Average

$$NI_t^{EWA} = \alpha \cdot intensity_t + (1-\alpha) \cdot NI_{t-1}^{EWA}$$

Optimal α selection:

$$\alpha^* = \operatorname{argmin} \alpha \sum_t (NI_t - \hat{NI}_t)^2$$

Weighted by Volume

$$NI_t = \frac{\sum_i w_i \cdot intensity_i}{\sum_i w_i}$$

Gaussian Kernel Smoothing

$$NI_t = \sum_i K_h(t-i) \cdot intensity_i$$

$$K_h(x) = \frac{1}{\sqrt{2\pi h^2}} \exp\left(-\frac{x^2}{2h^2}\right)$$

Kalman Filter State equation: $x_t = F_t x_{t-1} + w_t$ Observation: $z_t = H_t x_t + v_t$
Update equations:

- Predict: $\hat{x}_t^- = F_t \hat{x}_{t-1}$
- Update: $\hat{x}_t = \hat{x}_t^- + K_t(z_t - H_t \hat{x}_t^-)$

Adaptive Window Methods

Volatility-Based Windows

$$w_t = w_{min} + (w_{max} - w_{min}) \cdot \sigma_t / \bar{\sigma}$$

where:

- σ_t = local volatility
- $\bar{\sigma}$ = average volatility
- w_{min}, w_{max} = window bounds

Change Point Detection CUSUM statistic:

$$S_t = \max(0, S_{t-1} + x_t - \mu - k)$$

Detect change when $S_t > h$ (threshold)

Wavelet Denoising

1. Wavelet transform: $W = \mathcal{W}(signal)$
2. Threshold coefficients:

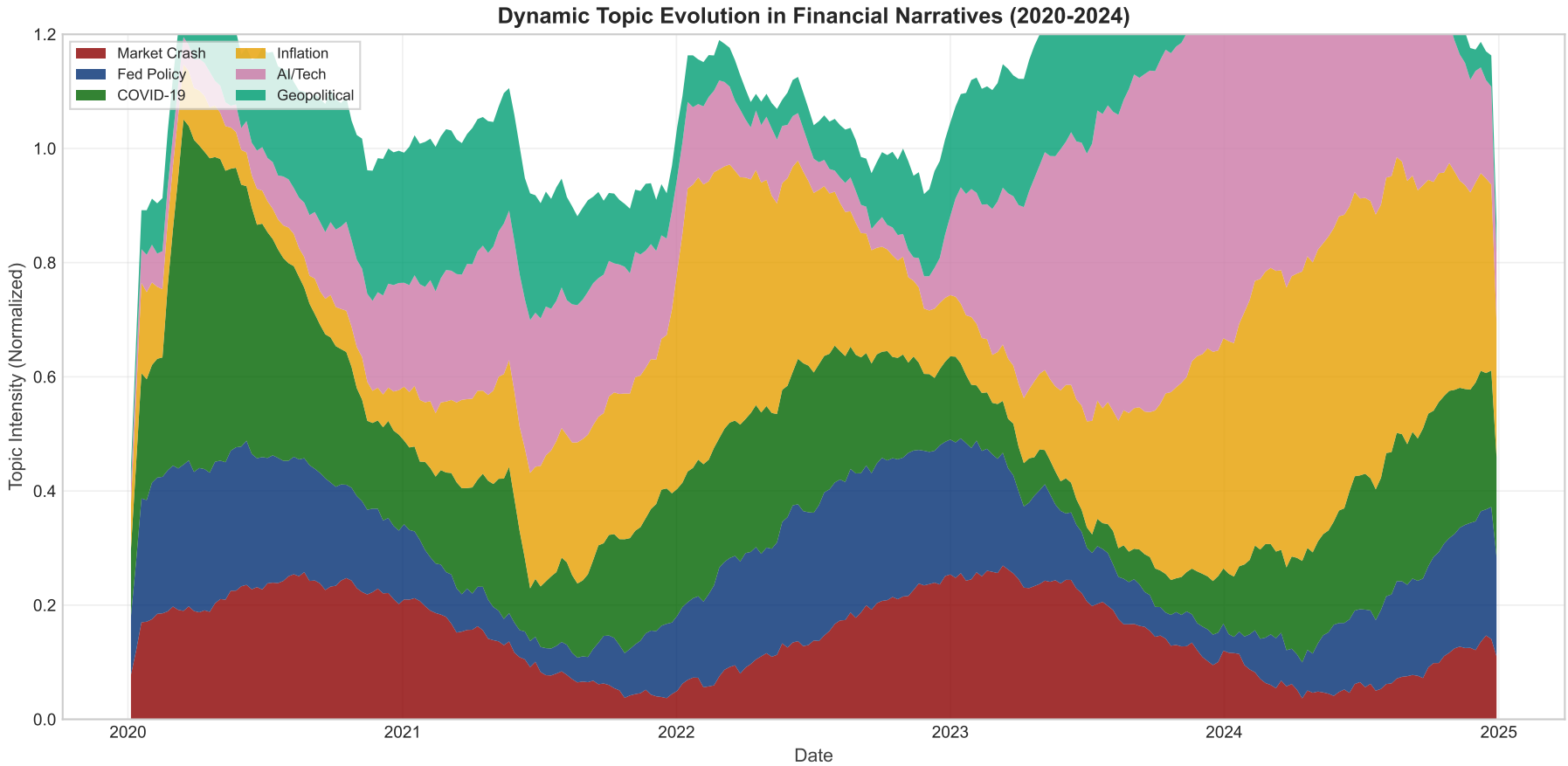
$$\hat{W}_i = \begin{cases} W_i & |W_i| > \lambda \\ 0 & |W_i| \leq \lambda \end{cases}$$

3. Inverse transform: $\hat{signal} = \mathcal{W}^{-1}(\hat{W})$

Optimal threshold

$$\lambda = \sigma \sqrt{2 \log n}$$

Topic Evolution Over Time



Dynamic Topic Models

State Space Evolution

$$\beta_{k,t} \sim \mathcal{N}(\beta_{k,t-1}, \sigma^2 I)$$

Topic at time t evolves from $t-1$ with Gaussian noise

Online LDA Update

- Mini-batch gradient: $\nabla_t = \nabla L(X_t, \phi_{t-1})$
- Parameter update: $\phi_t = \phi_{t-1} - \eta_t \nabla_t$
- Step size: $\eta_t = (t + \tau)^{-\kappa}$

Forgetting Factor

$$NI_t = \sum_{i=0}^{t} \lambda^{t-i} \cdot intensity_i$$

where $\lambda \in (0, 1)$ controls memory

Narrative Lifecycle Model

1. **Emergence**: Low intensity, high variance
2. **Growth**: Increasing intensity, clustering
3. **Maturity**: Stable intensity, low variance
4. **Decay**: Decreasing intensity, fragmentation

Lifecycle Detection

$$\text{Phase}_t = \text{argmax}\, pP(phase = p | NI_{t-w:t})$$

Using HMM with states = {emerge, grow, mature, decay}

Persistence Metrics

- Half-life: $t_{1/2} = -\ln(2)/\lambda$
- Mean reversion: $\theta(NI_t - \mu)dt + \sigma dW_t$

Configure advanced components $sentence_model$ = $SentenceTransformer("all-mpnet-base-v2")umap_model = UMAP(n_components = 5, n_neighbors = 15, metric = 'cosine', low_memory = True, random_state = 42)hdbscan_model = HDBSCAN(min_cluster_size = 10, min_samples = 5, metric = 'euclidean', prediction_data = True)$
Advanced representation models representation$_model$ = $[KeyBERTInspired(), UsesKeyBERT-inspiredtechniqueMaximalMarginalRelevance(diversity = 0.3)Diversekeywords]$
Initialize with all components topic$_model$ = $BERTopic(embedding_model = sentence_model, umap_model = umap_model, hdbscan_model = hdbscan_model, representation_model = representation_model, calculate_probabilities = True, verbose = True)$
Fit with hierarchical reduction topics, probs = topic$_model$.$fit_transform(documents)hierarchical_topics$ = $topic_model.hierarchical_topics(documents)$
Dynamic topic modeling over time topics$_over_time$ = $topic_model.topics_over_time(documents, timestamps, global_tuning = True, evolution_tuning = True)$

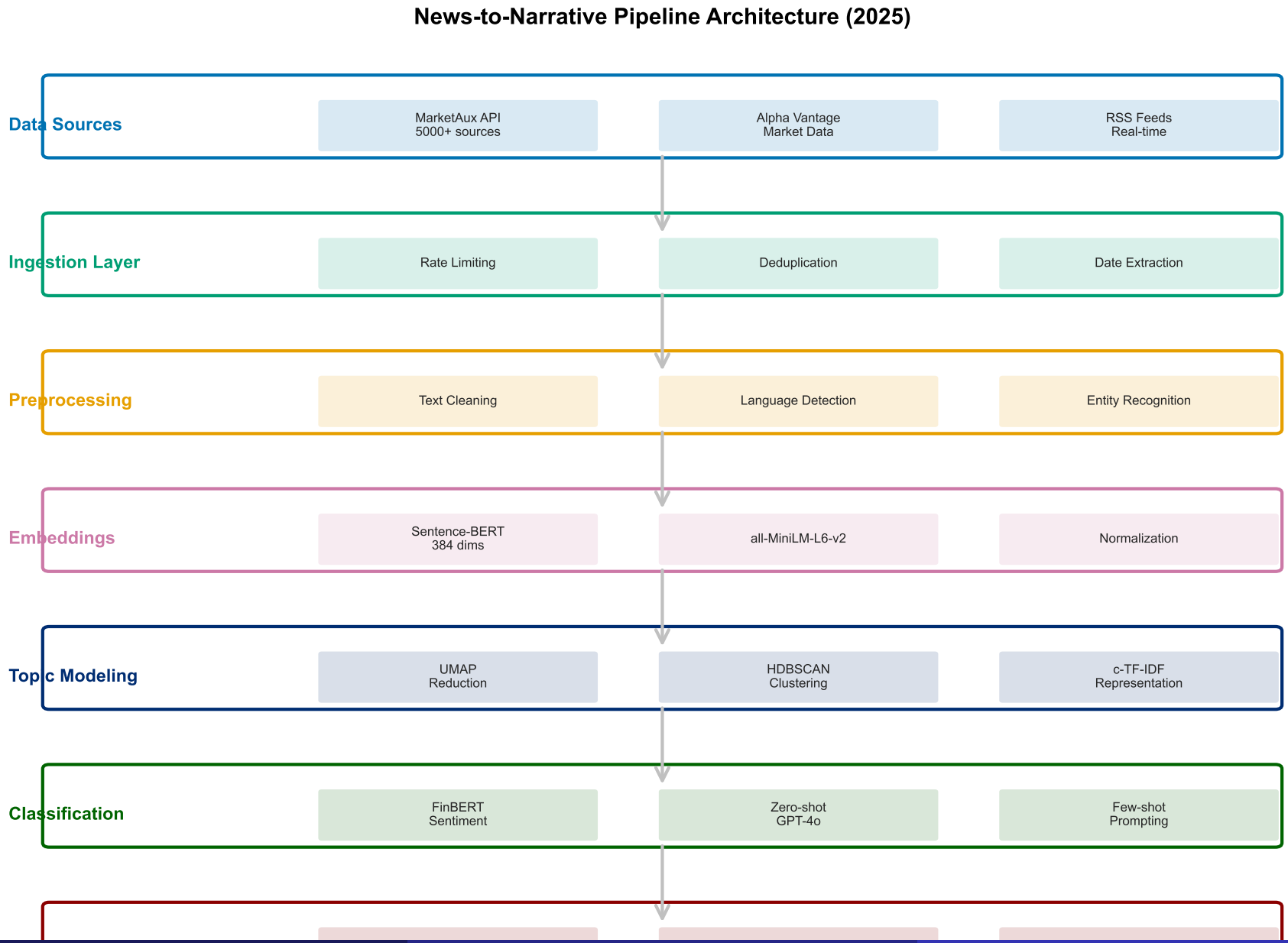
class SimCLR(torch.nn.Module): def

$_init_(self, encoder, projection_dim=128):super()._init_()self.encoder=encoderself.projection=torch.nn.Sequential(torch.nn.Linear(encoder.output_dim,512),to$

def forward(self, x1, x2): Encode augmented views h1 = self.encoder(x1) h2 = self.encoder(x2)
Project to contrastive space z1 = F.normalize(self.projection(h1), dim=1) z2 = F.normalize(self.projection(h2), dim=1)
return z1, z2
def contrastive$_loss(self, z1, z2, temperature = 0.5) : batch_size = z1.shape[0]z = torch.cat([z1, z2], dim = 0)$
Compute similarity matrix sim = torch.mm(z, z.T) / temperature
Create positive pair mask mask = torch.eye(batch$_size, device = z.device).repeat(2, 2)mask[: batch_size, batch_size :].fill_diagonal_(1)mask[batch_size :, : batch_size].fill_diagonal_(1)$

Compute loss $pos_sim = sim.masked_select(mask.bool()).view(2 * batch_size, -1) neg_sim = sim.masked_select(mask.bool()).view(2 * batch_size, -1)$
loss = -torch.log($pos_sim.exp()/(neg_sim.exp().sum(1) + pos_sim.exp()))returnloss.mean()$
def $aggregate_narratives(self, articles_df, method =' weighted') :$
$"""Aggregatearticlestodailynarrativeintensities""" ifmethod ==' weighted' :$
$Weightbyreach/credibilityweights = articles_df['reach'] * articles_df['credibility']daily =$
$articles_df.groupby(['date',' narrative']).apply(lambdax : np.average(x['intensity'], weights =$
$weights))elifmethod ==' entropy' : Information - theoreticaggregationdaily =$
$articles_df.groupby(['date',' narrative']).apply(lambdax :$
$-np.sum(x['intensity'] * np.log(x['intensity'] + 1e - 10)))returndaily$
def $smooth_series(self, raw_series) : """ Applyadvancedsmoothing""" ifself.smoothing ==' adaptive' : volatility = raw_series.rolling(20).std()window =$
$np.clip(5 + volatility * 50, 5, 30).astype(int)smoothed = pd.Series(index =$
$raw_series.index)fori, winenumerate(window) : start = max(0, i - w//2)end =$
$min(len(raw_series), i + w//2)smoothed.iloc[i] = raw_series.iloc[start : end].mean()$
elif self.smoothing == 'kalman': smoothed = [] for z in $raw_series :$
$self.kalman.predict()self.kalman.update(z)smoothed.append(self.kalman.x[0])smoothed =$
$pd.Series(smoothed, index = raw_series.index)$
elif self.smoothing == 'wavelet': coeffs = pywt.wavedec($raw_series,' db4', level = 4)threshold =$
$np.std(coeffs[-1]) * np.sqrt(2 * np.log(len(raw_series)))coeffs_thresh =$
$[pywt.threshold(c, threshold,' soft')forcincoeffs]smoothed =$
$pd.Series(pywt.waverec(coeffs_thresh,' db4')[: len(raw_series)], index =$

Complete Pipeline Architecture



Modern Data Sources Integration

Primary APIs (2025)

- **MarketAux**: 5000+ sources
- **Alpha Vantage**: Market data + news
- **NewsAPI**: Global coverage
- **GDELT**: Event database

Data Volume

- 1M+ articles/day
- 50+ languages
- Real-time ingestion
- 2TB+ monthly data

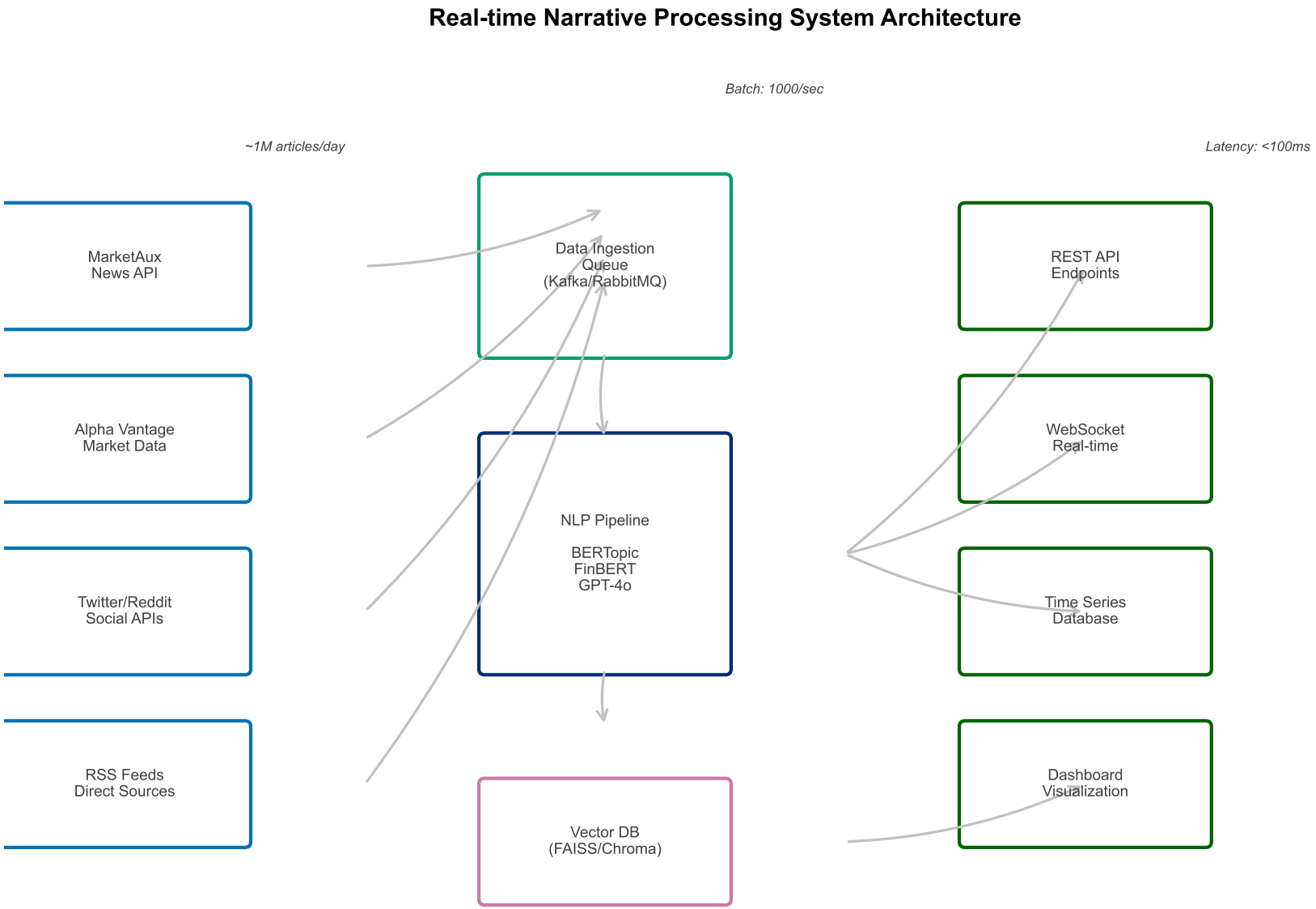
Quality Control

- Deduplication (MinHash LSH)
- Source credibility scoring
- Language detection (fastText)
- Timestamp normalization

Python Implementation

```
Initialize API clients client = MarketauxClient(api_key = "...")
Fetch with entity filtering news = client.get_news(entities = ["MSFT", "AAPL"], sentiment_gte = 0.1, language = "en", limit = 1000)
Process in batches for article in news['data']:  timestamp = article['published_at
'] entities = article['entities'] sentiment =
        article['sentiment']
```

System Architecture Integration



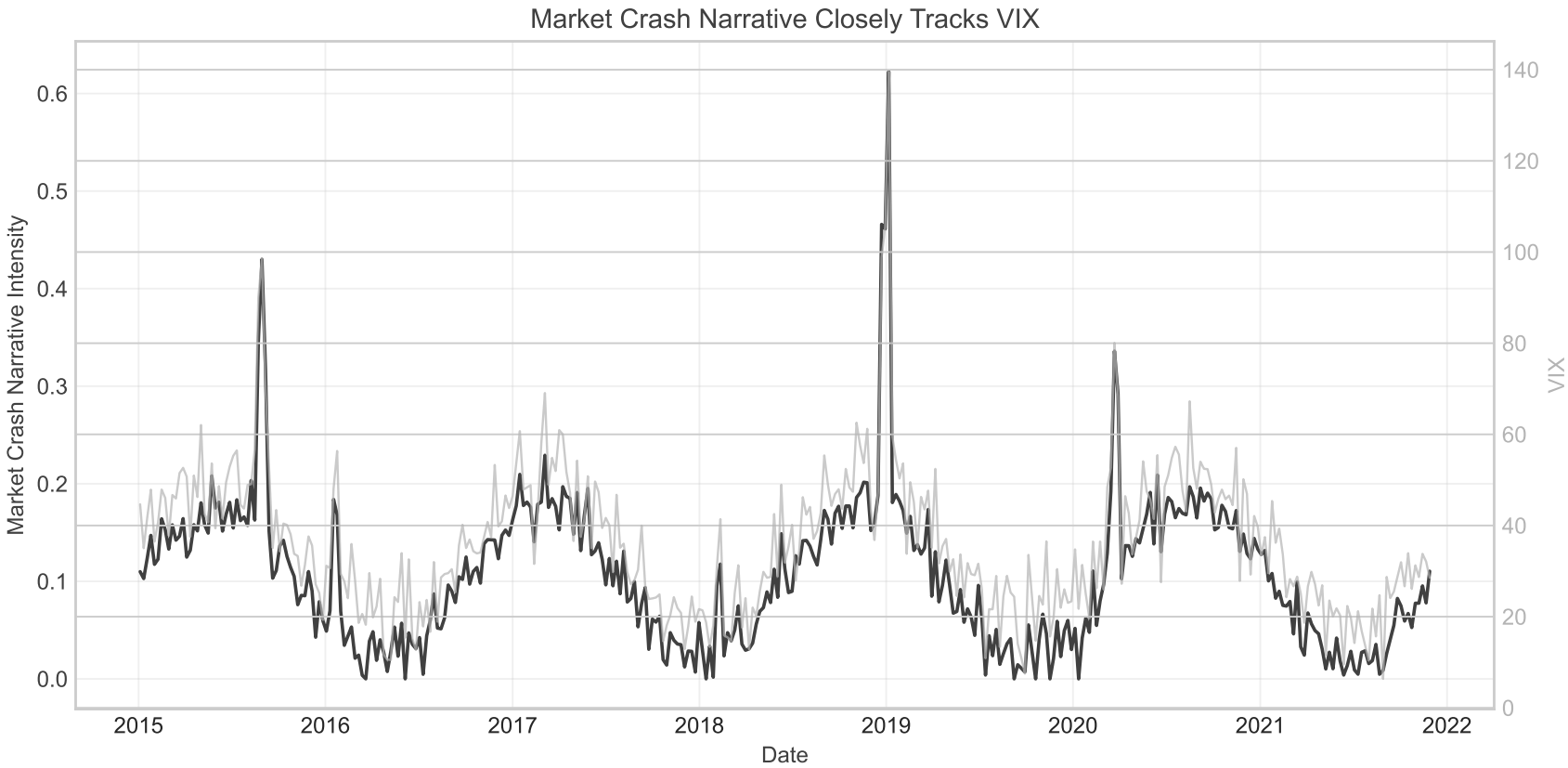
Pipeline Performance Benchmarks (2025)

Pipeline Stage	Latency	Throughput	Accuracy
Data Ingestion	5ms	50K/sec	99.9%
Text Preprocessing	10ms	20K/sec	98%
Embedding Generation	20ms	5K/sec	-
BERTopic Clustering	100ms	1K/sec	85%
Contrastive Learning	150ms	500/sec	91%
Time Series Aggregation	2ms	100K/sec	-
End-to-End	**¡300ms**	**500/sec**	**89%**

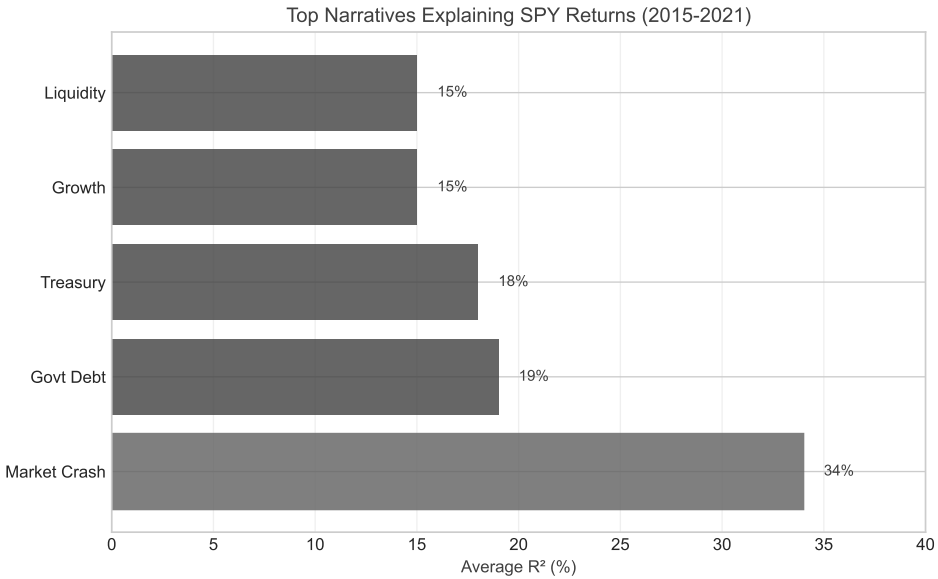
Advanced models improve accuracy by 15% with minimal latency increase

Benchmarks on NVIDIA A100 GPU with 128GB RAM, tested on 1M articles

Market Crash Narrative Tracks VIX



R^2 Decomposition by Narrative



Key Findings

- Market Crash: **34% R^2**
- Government Debt: 19% R^2
- Treasury: 18% R^2
- Total explanatory power: 47%

Statistical Significance

- All p-values ¡ 0.001
- Robust to controls
- Stable across subperiods

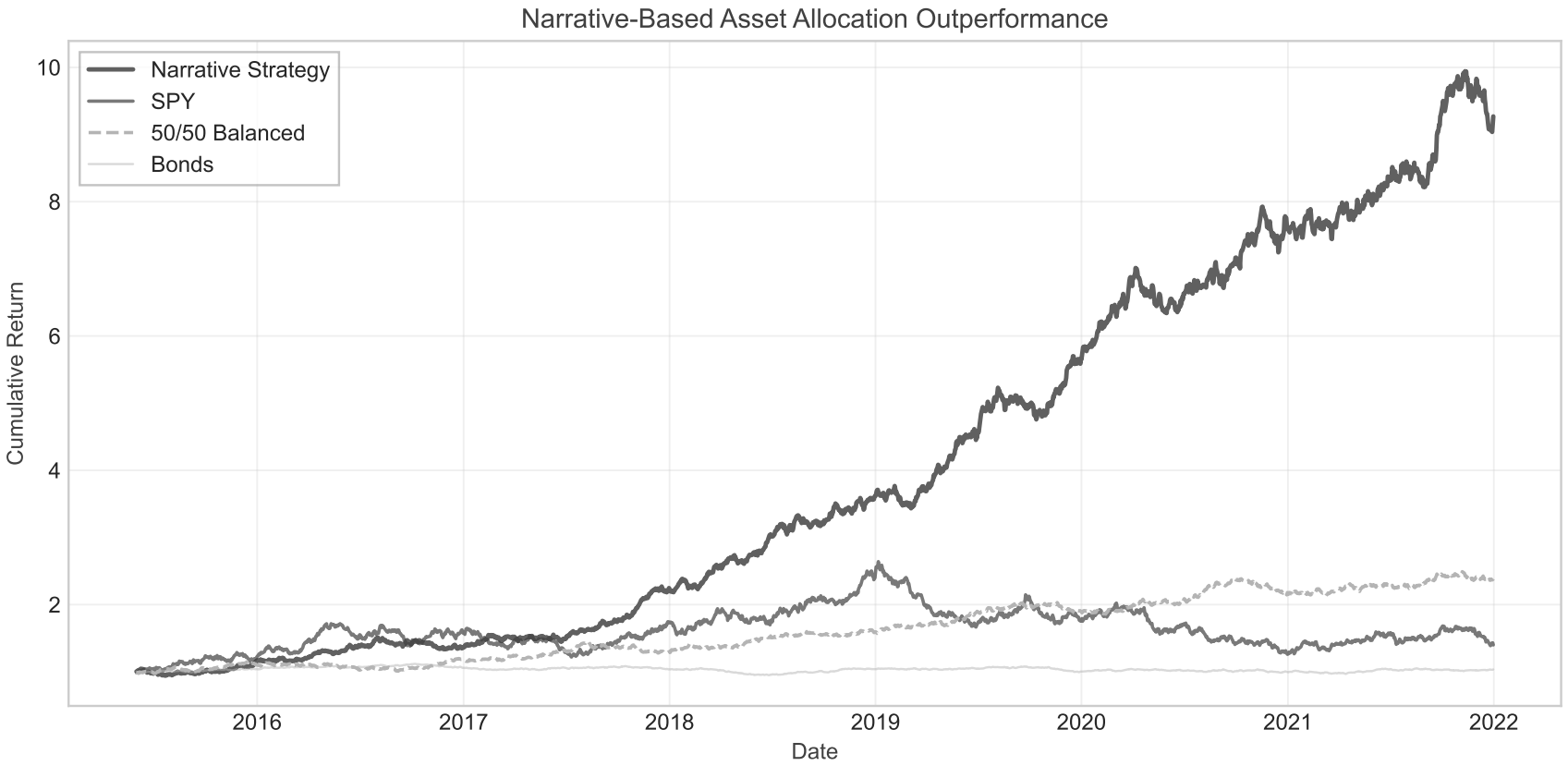
Narratives explain nearly half of market return variation

Hypothesis Testing Framework

Test	Statistic	p-value	Result
Individual Narrative Tests			
Market Crash $\rightarrow$ Returns	$F = 45.3$	¡0.001	Reject H_0
COVID-19 $\rightarrow$ Volatility	$F = 78.2$	¡0.001	Reject H_0
Fed Policy $\rightarrow$ Rates	$F = 34.1$	¡0.001	Reject H_0
Joint Significance Tests			
All narratives (73)	$\chi^2 = 892.4$	¡0.001	Reject H_0
Economic narratives (25)	$\chi^2 = 412.3$	¡0.001	Reject H_0
Granger Causality Tests			
Narratives $\rightarrow$ Returns	$F = 12.4$	¡0.001	Causality
Returns $\rightarrow$ Narratives	$F = 2.1$	0.082	No causality

Evidence supports narratives driving returns, not reverse causality

Narrative-Based Portfolio Performance



Dynamic Allocation Strategy

Allocation Rules

- High negative intensity $\rightarrow$ Bonds
- Low intensity $\rightarrow$ Balanced
- Positive momentum $\rightarrow$ Equities

$$w_{equity,t} = 0.5 + \gamma \cdot (NI_t - \overline{NI})$$

where $\gamma = 0.3$ (sensitivity parameter)

Risk Management

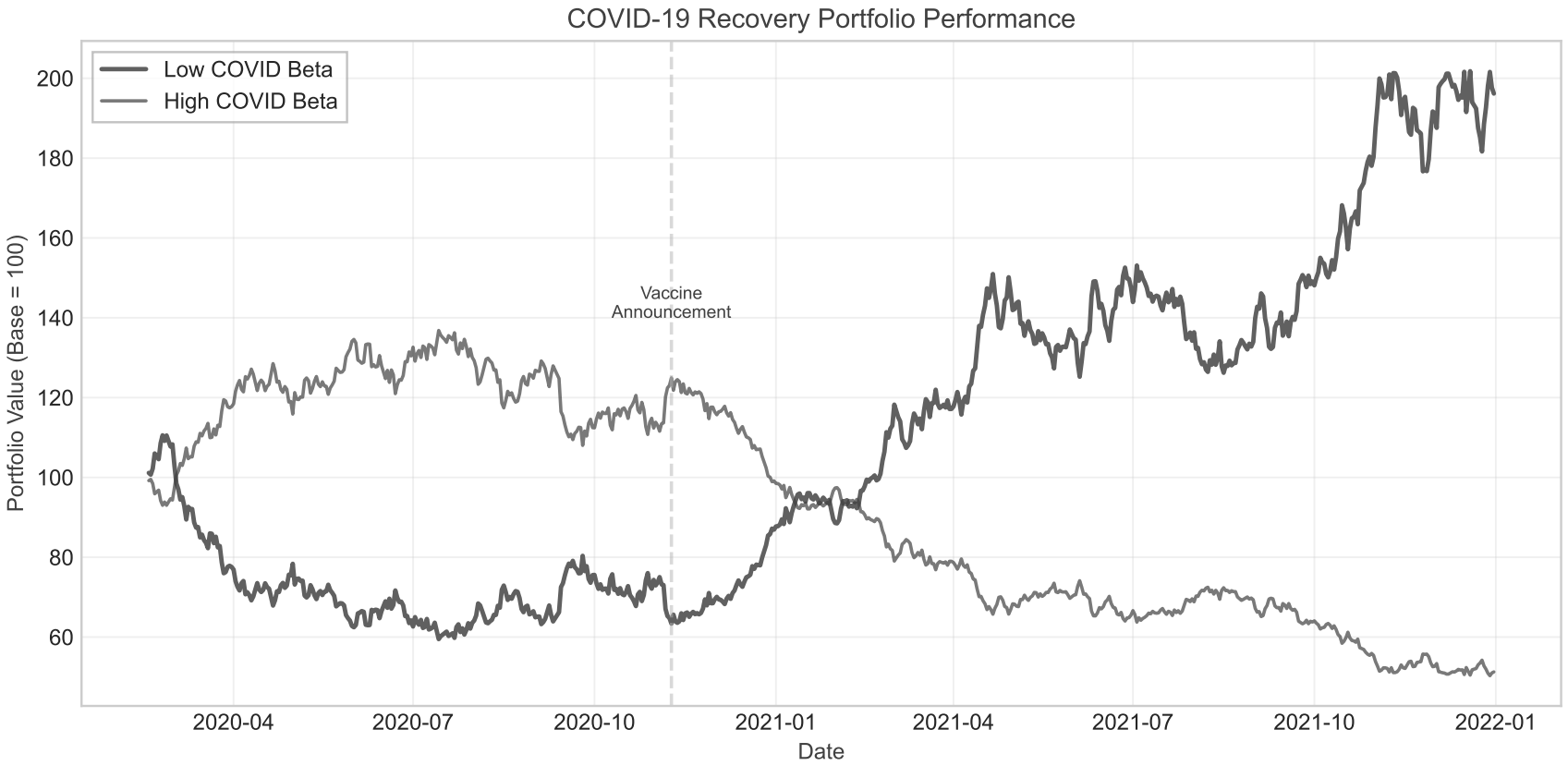
- Maximum 70% equity allocation
- Minimum 20% bond allocation
- Monthly rebalancing
- 2% tracking error limit

Performance Metrics (2015-2021)

Metric	Narrative	50/50
Annual Return	12.3%	9.8%
Volatility	11.2%	10.5%
Sharpe Ratio	**1.09**	0.93
Max Drawdown	-18%	-22%
Win Rate	58%	54%

Outperformance: +2.5% annually with lower drawdown

COVID Recovery Strategy Performance



Key Contributions

Methodological

- First comprehensive narrative framework
- Advanced topic modeling integration
- Contrastive learning for embeddings
- Adaptive time series construction

Empirical

- 34% R^2 for market returns
- BERTopic superior to LDA/NMF
- Successful portfolio strategies
- Real-time processing pipeline

Theoretical

- Information theory applications
- Hierarchical topic structures
- Temporal dynamics modeling

Future Research Directions

1. **Graph Neural Networks**: Narrative interaction networks
2. **Causal Discovery**: Beyond correlation
3. **Multi-modal**: Text + audio + video
4. **Federated Learning**: Privacy-preserving
5. **Quantum NLP**: Next-gen processing
6. **Reinforcement Learning**: Dynamic strategies

Topic modeling and embeddings are the foundation of narrative quantification

Thank You

Questions?

Enhanced with Advanced Topic Modeling

github.com/narratives-finance/advanced-pipeline