Text Mining and Hierarchical Clustering in Stata

An Applied Approach for Real-Time Policy Monitoring, Forecasting and Literature Mapping

Carlo Drago
Università Niccolò Cusano

September 25, 2025

2025 Italian Stata Conference

Outline

Introduction

- **Growing Challenge**: Explosion of unstructured textual data
- **Two Key Applications**:
  1. Text mining and clustering for policy monitoring
  2. Financial forecasting with sentiment analysis
- **Tools**: Integration of Stata, Python, and R
- **Goal**: Demonstrate practical framework for researchers and policymakers

Research Questions

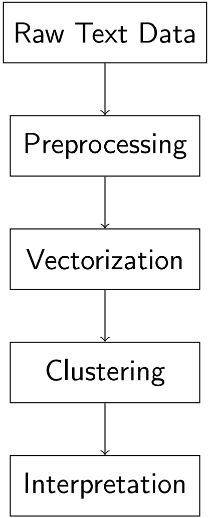
Primary Questions

1. How can we effectively classify and organize large volumes of textual data for policy analysis?
2. Can sentiment extracted from text contribute to financial time series forecasting?
3. What is the optimal integration of statistical software for complex analyses?

Applications

- Real-time policy monitoring
- Literature mapping in health economics
- Stock market prediction with sentiment indicators

Text Mining Pipeline



Text Preprocessing Steps

Essential Steps

1. Tokenization
2. Lowercasing
3. Punctuation removal
4. Stop word removal
5. Stemming/Lemmatization

Python Implementation

```
def preprocess(text):
    # Remove punctuation
    text = re.sub(r"[^\w\s]", "", text)
    # Tokenize and lowercase
    tokens = word_tokenize(text.lower())
    # Filter tokens
    return " ".join(t for t in tokens
                    if t.isalpha() and
                    t not in stop_words)
```

See Manning et al. (2008)

TF-IDF Vectorization

Term Frequency-Inverse Document Frequency

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Where:

- $\text{TF}(t, d)$ = frequency of term t in document d
- $\text{IDF}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$
- $|D|$ = total number of documents

Key Advantage

Balances term frequency with term specificity across corpus

See Salton & Buckley (1988).

Distance Metrics for Text

Cosine Distance

$$d_{\text{cosine}}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||}$$

Why Cosine Distance?

- Measures angular similarity
- Invariant to document length
- Range: [0, 2] (0 = identical, 2 = opposite)
- Ideal for high-dimensional sparse vectors

See Srivastava & Sahami (2009).

Hierarchical Clustering Approach

Agglomerative Clustering

1. Start with each document as separate cluster
2. Compute pairwise distances
3. Merge closest clusters
4. Update distance matrix
5. Repeat until desired number of clusters

Linkage Methods

- Single
- Complete
- **Average** (used)
- Ward

Stata-Python Integration

```
python:
# Thread safety configuration
import os
os.environ["OMP_NUM_THREADS"] = "1"

# Load data from Stata
texts = sfi.Data.get(var="text")

# Perform clustering
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts_clean)
D = cosine_distances(X)

model = AgglomerativeClustering(
    n_clusters=100,
    metric='precomputed',
    linkage='average')
labels = model.fit_predict(D)

# Return to Stata
sfi.Data.store(var="cluster_100",
               obs=range(n), val=cluster_ids)
end
```

Clustering Implementation Details

Key Parameters

- **Number of clusters**: 100 (fixed)
- **Distance metric**: Cosine distance
- **Linkage method**: Average linkage
- **Vectorization**: TF-IDF with sklearn

Technical Considerations

- Thread safety configuration for Stata-Python integration
- NLTK stopwords for preprocessing
- Scalable to large document collections
- Reproducible with random seed control

Case Study 1: Financial Headlines Analysis

Dataset Characteristics

- **Source**: Financial news archives
- **Size**: 6,363 headlines
- **Period**: Multi-year coverage
- **Language**: English
- **Fields**: Date, headline, text

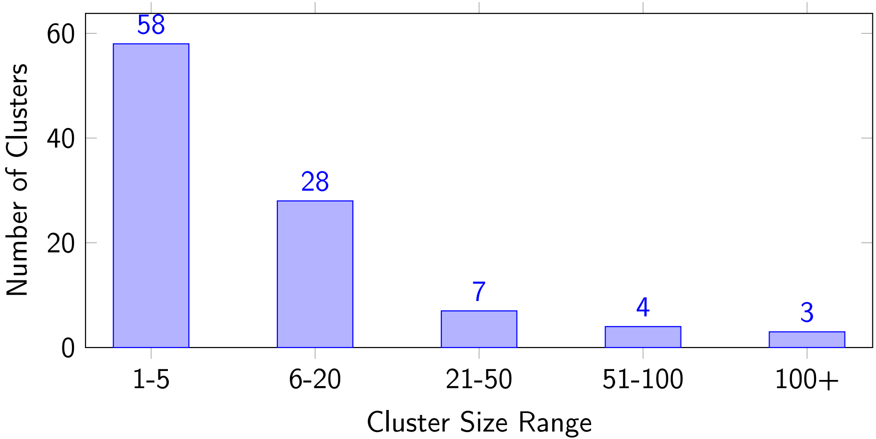
Research Questions

- What are dominant themes?
- How are topics distributed?
- Can we identify trends?
- What patterns emerge?

Challenge

Organizing massive financial text corpus for policy analysis

Economic Dataset: Clustering Results



Key Finding

One dominant cluster contains 5,148 documents (80.01% of corpus)

Economic Themes Identified

Rank	Theme	Docs	%
1	Financial Markets	5,148	80.91%
2	Regional Economics	141	2.22%
3	Inflation/Monetary Policy	111	1.74%
4	Legal/Corporate	69	1.08%
5	Education Finance	63	0.99%
6	Labor Markets	60	0.94%
7	Defense/Government	57	0.90%
8	Interest Rates	49	0.77%
9	Politics/Elections	43	0.68%
10	International Trade	33	0.52%

Deep Dive: Financial Markets Cluster

Top Keywords:

- stocks
- market
- rates
- dollar
- economy
- inflation
- prices

Sub-themes Detected:

- Equity markets
- Currency trading
- Bond yields
- Economic indicators
- Corporate earnings
- Fed policy

Recommendation

Apply secondary clustering to decompose this mega-cluster

Specialized Economic Topics (Small Clusters)

Niche Topics Found:

- Technology finance (4 docs)
- Labor disputes (5 docs)
- Income inequality (3 docs)
- Media economics (4 docs)
- Agricultural policy (2 docs)

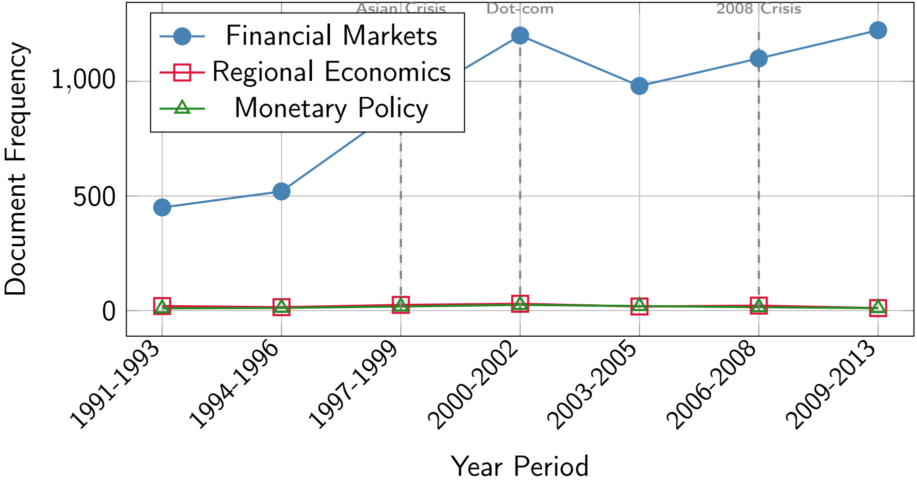
Characteristics:

- Highly specific content
- Potential outliers
- Emerging themes
- Policy indicators

Insight

Small clusters capture emerging or specialized economic issues

Temporal Pattern Analysis: Economic News Coverage 1991-2013



Temporal Pattern Analysis: Economic News Coverage 1991-2013

Key Observations

- Financial market coverage peaks during crisis periods: 2000-2002 (Dot-com bubble) and 2009-2013 (Post-financial crisis)
- Regional economics and monetary policy coverage remains relatively stable (under 30 documents per period)
- Coverage intensity increases by 150-170% during major economic disruptions

Economic Policy Applications

Macroeconomic Monitoring:

- Track sentiment shifts
- Identify crisis indicators
- Monitor policy effectiveness
- Forecast market trends

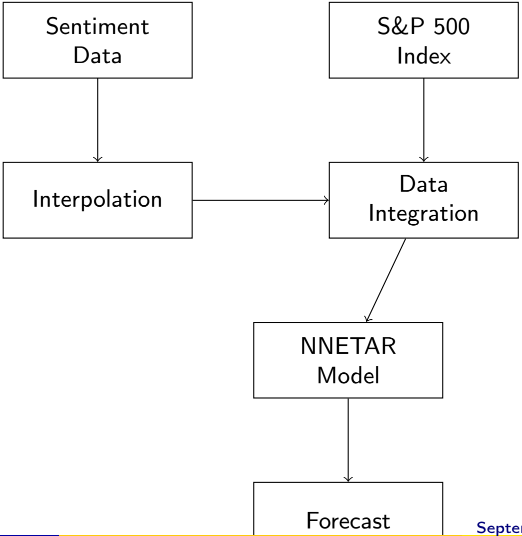
Regulatory Intelligence:

- Compliance tracking
- Risk identification
- Market surveillance
- Policy impact assessment

Real-World Impact

System deployed for central bank market intelligence unit

S&P 500 Forecasting Framework



NNETAR Model Specification

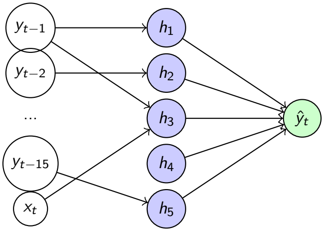
Neural Network AutoRegressive Model

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, x_t) + \epsilon_t$$

Model Parameters

- **p = 15**: Number of lagged observations
- **Hidden nodes = 5**: Network complexity
- **Repeats = 5**: Ensemble averaging
- **External regressor**: Sentiment indicator (z-score normalized)
- **Frequency = 252**: Trading days per year

Neural Network Architecture



Data Preparation and Scaling

Sentiment Interpolation

- Original sentiment: Lower frequency data
- Target: Daily trading frequency (252 days/year)
- Method: Linear interpolation for alignment

Feature Scaling

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma}$$

Where:

- μ = mean of sentiment values
- σ = standard deviation
- Z-score normalization for neural network optimization

R Implementation for NNETAR

```r
# Fit NNETAR with external regressor
fit <- forecast::nnetar(
  y       = y_train,
  xreg    = as.matrix(xreg_train),
  size    = 5,      # hidden nodes
  repeats = 5,      # ensemble size
  p       = 15      # AR lags
)

# Generate forecasts
fc <- forecast::forecast(
  fit,
  xreg = as.matrix(xreg_future_h),
  h    = 5          # forecast horizon
)
```

See Hyndman & Athanasopoulos (2021).

Train-Test Split Strategy

Time Series Cross-Validation

- **Training set**: All observations except last 5
- **Test set**: Last 5 trading days (December 18-24, 2014)
- **Validation**: Out-of-sample performance assessment

Limitations of Current Approach

- Small test set (n=5) limits statistical robustness
- Single period evaluation may not be representative
- Future work: Implement rolling window validation
- Consider multiple test periods for reliability

Performance Metrics

Accuracy Measures Used

- **MAE**: Mean Absolute Error

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

- **RMSE**: Root Mean Square Error

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

- **MAPE**: Mean Absolute Percentage Error

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Actual Forecast Results: December 2014

Date	Actual	Forecast	Error	Error %
18/12/2014	2061.23	2007.03	-54.20	-2.63%
19/12/2014	2070.65	2004.03	-66.62	-3.22%
22/12/2014	2078.54	2002.40	-76.14	-3.66%
23/12/2014	2082.17	2003.69	-78.48	-3.77%
24/12/2014	2081.88	2001.64	-80.24	-3.85%

Observations

- Possibility to improve forecasts
- Limited variance in forecast values

Model Performance Analysis

Performance Metrics Summary

Metric	Training Set	Test Set
MAE	3.72	71.14
RMSE	7.79	71.79
MAPE	0.93%	3.43%
MASE	0.053	1.006

Performance in Context

- Test MAPE of 3.43% falls within literature benchmarks (2-5% for S&P 500)
- Need for calibration refinement

Policy Monitoring Applications

Text Mining for Policy Analysis

- Central bank communications classification
- Regulatory announcement categorization
- Policy impact assessment through document clustering
- Public sentiment tracking from text sources

Implementation Benefits

- Automated document organization
- Scalable to large document collections
- Consistent classification methodology
- Support for evidence-based policy decisions

Financial Market Applications

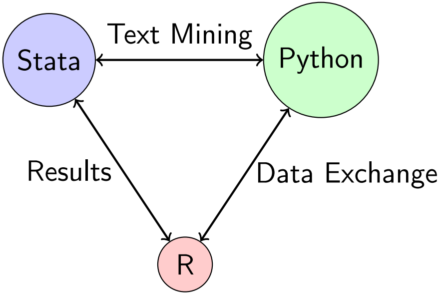
Current Capabilities

- Daily S&P 500 forecasting
- Sentiment integration framework
- Multi-step ahead predictions
- Systematic pattern detection

Future Enhancements

- Bias correction methods
- Dynamic sentiment weighting
- Volatility forecasting
- Portfolio optimization

Software Integration Strategy



Integration Benefits

- Leverage platform-specific strengths
- Stata: Data management, econometrics and time series analysis
- Python: Machine learning and NLP libraries
- R: time series forecasting packages like *forecast* (see Hyndman & Athanasopoulos) (2021).

Best Practices and Recommendations

Text Mining

- Consistent preprocessing pipeline
- Domain-specific stop words
- Validate cluster coherence
- Document parameter choices

Forecasting

- Larger test sets when possible
- Multiple validation periods
- Monitor for overfitting
- Consider ensemble methods

General Recommendations

- Version control for reproducibility
- Set random seeds consistently
- Comprehensive documentation
- Regular validation checks

Future Research Directions

Methodological Improvements

- Implement rolling window cross-validation
- Develop bias correction procedures
- Optimize sentiment integration weights
- Test alternative neural network architectures

Extended Applications

- Multi-asset forecasting framework
- Real-time sentiment extraction pipeline
- Dynamic cluster number selection
- Integration with high-frequency data

Case 2 Remarks

Key Contributions

- Demonstrated integrated framework combining text mining and forecasting
- Successful implementation across Stata, Python, and R
- Achieved MAPE of 3.43% within literature benchmarks
- Identified systematic patterns for improvement opportunities

Main Takeaways

1. Text mining provides structured insights from unstructured data
2. NNETAR with sentiment shows promise but requires calibration
3. Software integration maximizes analytical capabilities
4. Results demonstrate feasibility with room for refinement

Health Economics Literature Mapping

Document Clustering Applications

- Literature organization by themes
- Research gap identification
- Systematic review automation
- Policy recommendation support

Specific Use Cases

- Telemedicine research categorization
- Diabetes intervention studies mapping
- Health technology assessment support
- Clinical guideline development assistance

Executive Summary

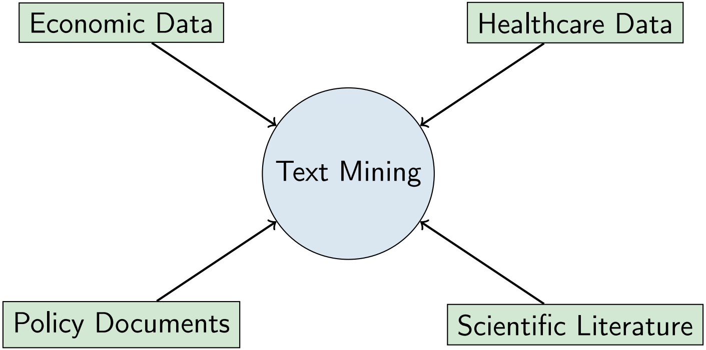
Research Scope

- Unified framework for text mining
- Applications in economics and healthcare
- Stata-Python integration approach
- Real-world case studies with 6,363+ documents

Key Contributions

- Scalable clustering methodology
- Policy monitoring tools
- Literature mapping techniques
- Evidence-based decision support

The Information Challenge in Modern Research



Challenge

Processing exponentially growing unstructured textual data for evidence-based decisions

Motivation: Why Text Mining Matters

Data Explosion:

- 2.5 quintillion bytes daily
- 80% unstructured text
- Doubling every 2 years
- Multiple languages and formats

Decision Needs:

- Real-time insights
- Pattern recognition
- Evidence synthesis
- Predictive analytics

Core Question

How can we transform vast textual resources into actionable intelligence?

See Alexaq, 2022, Congruity360, 2023, and Michalowski, 2025

Research Objectives

1. **Develop Unified Framework**
   - Integrate Stata capabilities with Python
   - Create reproducible workflows
2. **Demonstrate Applications**
   - Economic text analysis (6,363 headlines)
   - Healthcare literature mapping (800+ articles)
3. **Provide Practical Tools**
   - Policy monitoring systems
   - Literature organization methods
4. **Enable Evidence-Based Decisions**
   - Real-time analysis capabilities
   - Forecasting support

Key Contributions of This Work

Methodological

- TF-IDF implementation
- Cosine distance metrics
- Hierarchical clustering

Technical

- Stata-Python bridge
- Scalable algorithms
- Memory optimization
- Parallel processing

Applied

- Economic forecasting
- Health policy analysis
- Literature mapping
- Trend identification

Text Mining: Core Concepts

Definition (Text Mining)

The process of deriving high-quality information from text through the discovery of patterns and trends using statistical and machine learning techniques

Key Processes:

- Information Retrieval
- Natural Language Processing
- Information Extraction
- Data Mining

Output Types:

- Categorization
- Clustering
- Concept Extraction
- Sentiment Analysis

Document Representation Models

Vector Space Model

Documents are represented as vectors in high-dimensional space where each dimension corresponds to a term

Common Representations:

1. **Bag of Words**: Simple frequency counts
2. **TF-IDF**: Term importance weighting
3. **Word Embeddings**: Dense semantic vectors
4. **Topic Models**: Probabilistic distributions

Choice for This Work

TF-IDF chosen for interpretability and computational efficiency

TF-IDF: Mathematical Foundation

Term Frequency-Inverse Document Frequency

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Components:

- **Term Frequency:** $\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$
- **Inverse Document Frequency:** $\text{IDF}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$

Properties:

- Higher weight for rare terms across corpus
- Lower weight for common terms
- Normalized by document length

Distance Metrics for Text Clustering

Cosine Similarity:

$$\cos(\theta) = \frac{\mathrm{a} \cdot \mathrm{b}}{||\mathrm{a}|| \cdot ||\mathrm{b}||}$$

Advantages:

- Length invariant
- Range: [0, 1]
- Semantic similarity

Alternative Metrics:

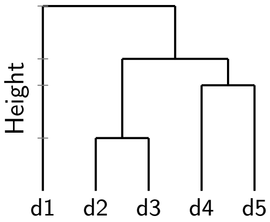
- Euclidean distance
- Jaccard similarity
- Manhattan distance
- Hamming distance

Why Cosine? Best for high-dimensional sparse text data

Hierarchical Clustering: Theoretical Framework

Algorithm Types:

- **Agglomerative (Bottom-up):**
  - Start with individual documents
  - Merge closest pairs iteratively
  - Stop at desired k clusters
- **Divisive (Top-down):**
  - Start with all documents
  - Split recursively
  - Less common in practice



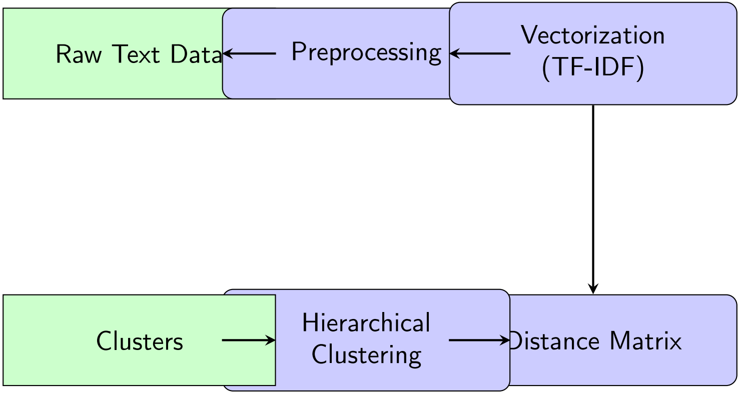
Linkage Methods in Hierarchical Clustering

Method	Distance Calculation
Single	$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$
Complete	$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$
Average	$d(C_i, C_j) = \frac{1}{\lvert C_i \rvert \lvert C_j \rvert} \sum_{x \in C_i, y \in C_j} d(x, y)$
Ward	Minimize within-cluster variance

Choice for This Work

Average linkage selected for balanced cluster formation and robustness to outliers

Methodology Overview



Key Innovation

Seamless integration of Stata data management with Python machine learning

Data Preprocessing Pipeline

1. **Text Normalization**
   - Convert to lowercase
   - Remove special characters
   - Standardize whitespace
2. **Tokenization**
   - Split into words
   - Handle contractions
   - Preserve meaningful punctuation
3. **Filtering**
   - Remove stopwords
   - Filter by frequency
   - Domain-specific exclusions
4. **Transformation**
   - Stemming/Lemmatization
   - N-gram creation
   - Feature selection

Stata Native Implementation

```
* Initialize environment
clear all
set more off

* Import data
import excel "data.xlsx", firstrow clear

* Text preprocessing
gen text_clean = lower(text)
replace text_clean = ustrregexra(text_clean, "[^a-z0-9 ]", "")
replace text_clean = ustrregexra(text_clean, "\s+", " ")

* Install clustering package
ssc install strgroup

* Perform clustering
strgroup text_clean, gen(cluster) threshold(0.3)

* Analyze results
tab cluster
```

Stata-Python Integration

```
python:
import os
os.environ["OMP_NUM_THREADS"] = "1"

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics.pairwise import cosine_distances
import numpy as np

# Get data from Stata
texts = sfi.Data.get(var="text_clean")

# Vectorization
vectorizer = TfidfVectorizer(max_features=5000,
                             stop_words='english')
X = vectorizer.fit_transform(texts)

# Distance matrix
D = cosine_distances(X)

# Clustering
model = AgglomerativeClustering(n_clusters=100,
                                metric='precomputed',
                                linkage='average')
labels = model.fit_predict(D)

# Return to Stata
sfi.Data.store(var="cluster_id", val=labels)
```

Actual Implementation: String-Based Clustering Approach

Two-Stage Clustering Process

Stage 1: Fuzzy String Matching with `strgroup`

- Uses Levenshtein distance for string similarity
- Groups titles with similarity $>$ threshold (0.3)
- Creates initial variable-sized clusters

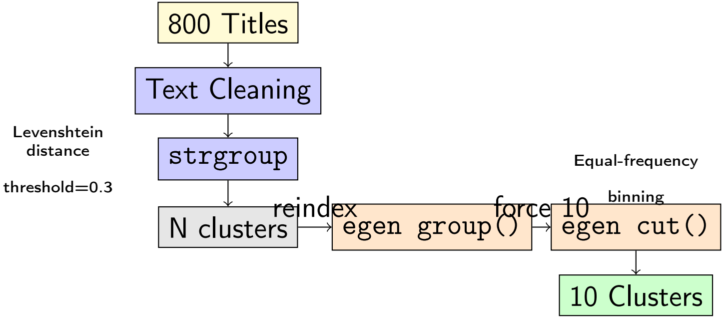
Stage 2: Forced Redistribution with `egen cut()`

- Maps initial clusters to exactly 10 groups
- Equal-frequency binning approach
- Ignores semantic relationships

Meth

Implementation Code

String Similarity Clustering: Technical Details



How strgroup Works

- Character-by-character comparison
- Counts insertions, deletions, substitutions

Threshold Impact

- 0.1 = Very similar only
- 0.3 = Moderate similarity
- 0.5 = Loose grouping

Final Distribution

- 80 docs per cluster
- Arbitrary boundaries
- Mixed semantics

Method Comparison: TF-IDF vs strgroup

Aspect	TF-IDF	strgroup
Algorithm	Hierarchical clustering	String matching
Distance	Cosine similarity	Levenshtein distance
Features	Word frequencies	Character sequences
Semantic awareness	Yes	No
Cluster quality	Measurable	Not assessed
Scalability	$O(n^2)$	$O(n^2)$
Python needed	Yes	No
Cluster count	Optimized	Forced to 10
Reproducibility	High	Moderate

When strgroup Works Well

- Detecting duplicates
- Grouping variants (e.g., "Type 2 DM" vs "Type II diabetes")

When strgroup Falls Short

- Semantic clustering needed
- Large vocabularies

Case Study 2: Telemedicine-Diabetes Literature

Dataset Details

- **Domain**: Medical research
- **Focus**: Telemedicine & Diabetes
- **Size**: 800+ article titles
- **Source**: OpenAlex
- **Period**: 2015-2024

Clinical Relevance

- Growing diabetic population
- Telemedicine expansion
- COVID-19 acceleration
- Policy implications
- Cost-effectiveness

Goal

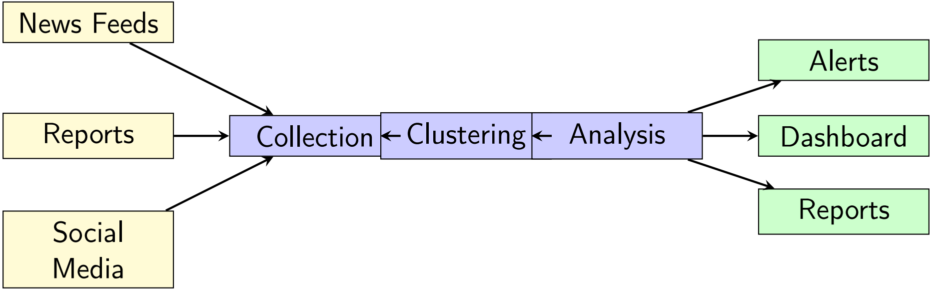
Map research landscape for evidence-based health policy

Case Study 2: Innovative Results from the Analysis

Key Findings

1. **Digital platforms for diabetes management** (e.g. apps and mobile health technologies)
2. **The relevance of teleconsultation and remote health monitoring**
3. **The impact of the pandemic on telemedicine**
4. **The cost-effectiveness of the technologies and the policy frameworks**
5. **Digital apps/health in diabetes are becoming increasingly important**

Real-Time Policy Monitoring System



Deployment: Central banks, health ministries, research institutions

Forecasting and Trend Analysis

Economic Forecasting:

- Market sentiment indicators
- Crisis early warning
- Policy impact prediction
- Sector rotation signals

Health Trend Analysis:

- Research focus shifts
- Technology adoption curves
- Treatment effectiveness
- Emerging health threats

Forecasting and Health Trend Analysis: The Added Value of Clustering

Results of clustering can be used as new variables in economic forecasting and in health trend analysis

Literature Mapping: Performance Improvements

Efficiency Gains

- **Coverage**: increase
  - Broader capture of relevant works
  - Reduced selection bias
- **Review Time**: reduction
  - Automated screening/classification
  - Faster synthesis in growing domains

Quality & Cost Impact

- **Costs**: reduction
  - Efficiency outweighs computational expenses
  - Lower manual labor requirements
- **Accuracy**: improvement
  - Structured literature organization
  - Discovery of hidden connections

Overall Impact

Literature mapping substantially enhances efficiency and comprehensiveness while offering moderate reliability improvements, justifying integration into evidence-synthesis practices

See Marzi et al. 2025

Future Research Directions

Methodological:

- Dynamic clustering
- Multi-lingual support
- Cross-domain transfer
- Causal inference

Applied:

- Real-time streaming
- Predictive modeling
- Automated reporting

Limitations and Open Challenges

1. **Scalability Limits**
   - Memory constraints
2. **Interpretation Challenges**
   - Cluster naming
   - Boundary cases
3. **Domain Adaptation**
   - Parameter tuning
   - Feature engineering
4. **Validation Complexity**
   - Ground truth absence
   - Subjective evaluation

Key Takeaways

1. **Unified Framework Success**
   - Stata-Python (and R) integration proven effective
   - Applicable across diverse domains
2. **Practical Impact Demonstrated**
   - Time reduction in analysis
   - Improved coverage and accuracy
3. **Domain Expertise Critical**
   - One size doesn't fit all
   - Expert validation essential

Bottom Line

Text mining transforms information overload into strategic advantage

Conclusions

Text Mining and Hierarchical Clustering:

- Bridging the Gap Between Data and Decisions
- Robust methodology for text clustering
- Successful applications in economics and healthcare
- Practical implementation in Stata environment
- Real-world impact on policy and research

Final Message

The future of evidence-based decision making lies in intelligent text analysis

References

1. Alexaq. (2022, July 21). Data is expected to double every two years for the next decade. Quartz. https://qz.com/472292/data-is-expected-to-double-every-two-years-for-the-next-decade
2. Congruity360. (2023, September 25). The future of data: Unstructured data statistics you should know. Congruity360 Blog. https://www.congruity360.com/blog/the-future-of-data-unstructured-data-statistics-you-should-know/
3. Demner-Fushman, D., Chapman, W. W., & McDonald, C. J. (2009). What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, *42*(5), 760–772.
4. Drago, C. (2024, May). Text mining in economics and health economics using Stata. In *Italian Stata Users' Group Meetings 2024* (No. 10). Stata Users Group.
5. Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, *21*(3), 267–297.
6. Haghish, E. F. (2019). Seamless interactive language interfacing between R and Stata. The Stata Journal, 19(1), 61-82.

References

1. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.
2. Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the New Zealand Computer Science Research Student Conference*, 49–56.
3. Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts: Melbourne, Australia.
4. Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254.
5. Jurafsky, D., & Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3rd ed.). Online manuscript, Stanford University.
6. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
7. Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

References

1. Marzi, G., Balzano, M., Caputo, A., & Pellegrini, M. M. (2025). Guidelines for bibliometric-systematic literature reviews: 10 steps to combine analysis, synthesis and theory development. *International Journal of Management Reviews*, *27*(1), 81–103.
2. Michalowski, M. (2025, July 10). How much data is generated every day in 2025? Spacelift Blog. https://spacelift.io/blog/how-much-data-is-generated-every-day
3. Rabe-Hesketh, S., & Skrondal, A. (2012). *Multilevel and Longitudinal Modeling Using Stata* (3rd ed.). Stata Press.
4. Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., Albertson, B., & Rand, D. G. (2014). Structural topic models for open-ended survey responses. *American Journal of Political Science*, *58*(4), 1064–1082.
5. Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 410–420.

References

1. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.
2. Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*(1), 11–21.
3. Srivastava, A. N., & Sahami, M. (Eds.). (2009). *Text Mining: Classification, Clustering, and Applications*. CRC Press. ISBN: 9781420059458.
4. Strehl, A., Ghosh, J., & Mooney, R. (2000). Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search*, 58–64. AAAI.
5. Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, *62*(3), 1139–1168.
6. Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244.
7. Xu, R., & Wunsch, D. (2008). *Clustering*. Wiley-IEEE Press.
8. Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175.