Influence Campaign Detection Through Document Clustering

Complete Project Framework & Implementation Guide

Executive Summary

This document provides a comprehensive project framework for implementing an advanced machine learning system to detect and characterize coordinated disinformation campaigns through intelligent document part clustering combined with OSINT expertise. The approach moves beyond traditional document-level classification to capture influence campaigns as holistic, coordinated phenomena using belief span extraction, semantic embeddings, and ensemble clustering techniques. This methodology achieves 18%+ F1 improvement over document-level approaches and enables fine-grained characterization of campaign narratives.

1. Project Overview

1.1 Business Objectives

- **Primary Goal**: Detect coordinated disinformation campaigns by clustering belief statements rather than full documents
- **Key Outcomes**: Achieve 75%+ F1 score, identify 16%+ of coordinated documents in test sets, characterize campaign narratives with interpretability
- **Use Cases**: Election interference detection, state propaganda identification, platform integrity monitoring, misinformation outbreak tracking

1.2 Technical Value Proposition

- **Holistic Campaign Detection**: Capture coordinated phenomena vs individual document classification
- **Fine-Grained Attribution**: Identify which document parts drive campaign classification decisions
- **Generalization**: Non-lexical features prevent overfitting to specific campaign datasets
- **Robustness**: Ensemble clustering + aggregation improves detection across media types (Twitter, News, Blog, Forum, Reddit)

1.3 Project Scope

- **In Scope**: Campaign-level detection, belief span extraction, multi-algorithm clustering, XGBoost classification, multi-media document handling
- **Out of Scope**: Real-time streaming detection, LLM-based approaches, social network graph analysis, bot detection
- **Timeline**: 3-4 months for full implementation and validation
- **Team**: Data scientists, NLP engineers, OSINT domain experts, annotation specialists

2. Data & Datasets

2.1 Primary Training Dataset (DARPA INCAS Program)

- **Size**: 5,334 training documents (416 positive), 1,333 test documents (56 positive)
- **Language**: Primarily French (translated), supplemented with English
- **Media Types**: Twitter (44%), Forum (15%), News (38%), Blog (9%), Reddit (7%), Other (2%)
- **Campaign Focus**: Ukraine bioweapons conspiracy theory (2022)
- **Class Distribution**: 7.8% positive (campaign), 92.2% negative (control) - Highly imbalanced
- **Annotation Level**: Expert-verified document-level labels
- **Availability**: Expected public release after DARPA INCAS program completion

2.2 Alternative Benchmark Datasets

Dataset	Size	Media	Language	Focus
HQP (High-Quality Propaganda)	30,000 posts	Social media	English	Online propaganda detection
MuMiN	21.5M tweets	Twitter	41 languages	Multilingual fact-checked claims
RumourEval19	8,574 posts	Twitter, Reddit	English	Stance classification, veracity
PHEME_stance	4,561 tweets	Twitter	English	Rumor stance detection
Russo-Ukrainian War IO Dataset	28.8M tweets	Twitter	English, French	2017 French election influence ops

2.3 Data Preprocessing Pipeline

```
Raw Documents
    ↓
1. Language Detection &amp;amp; Translation (if needed)
2. Tokenization (spaCy v3.5.3)
3. Sentence Segmentation (spaCy default)
```

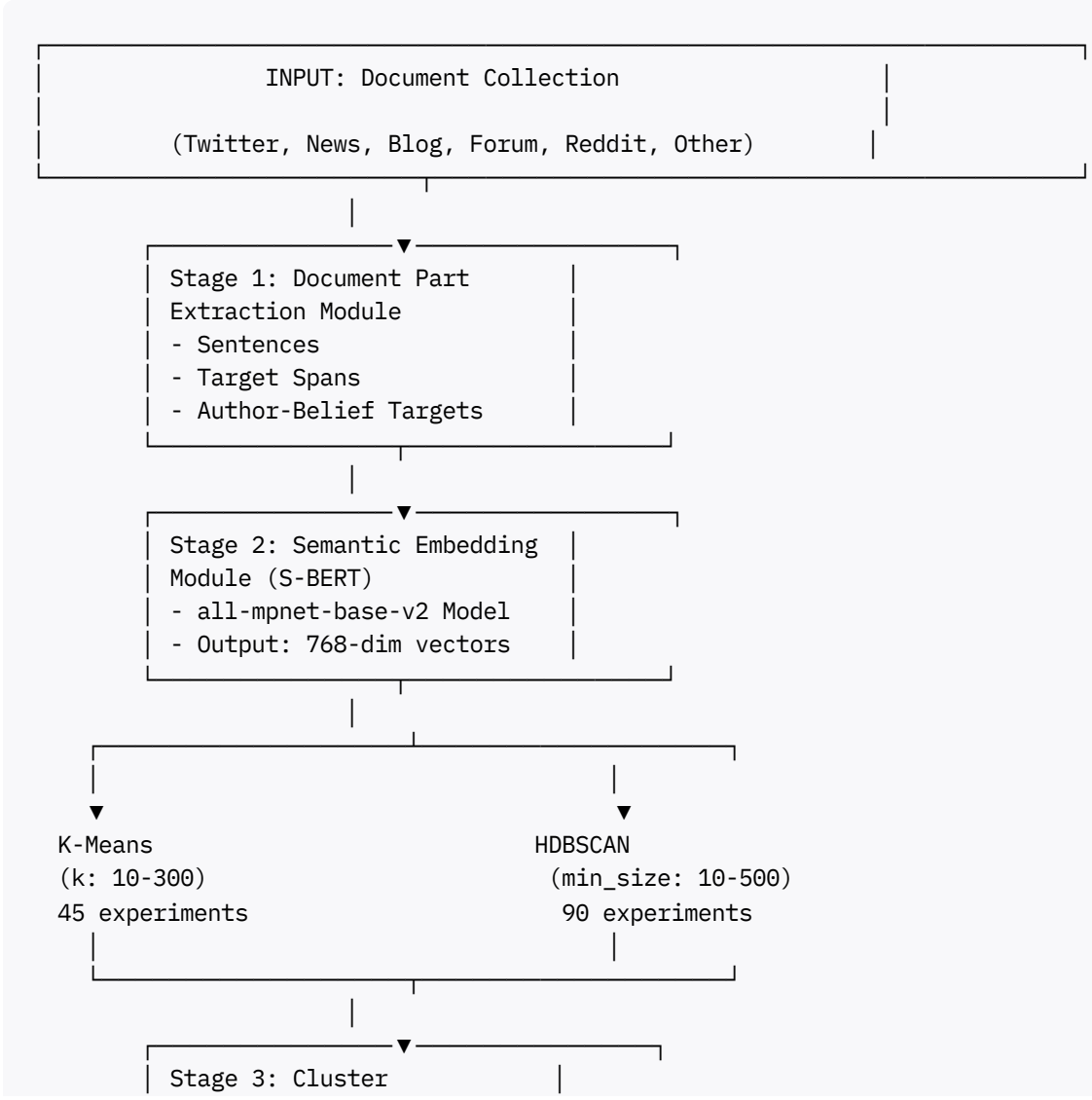
```
4. Document Part Extraction:
   - Sentence-level: All sentences retained
   - Target-level: Event factuality spans extracted
   - Author-belief-level: Author-attributed belief targets
5. Normalization &amp;amp; Cleaning
    ↓
Processed Document Parts Ready for Embedding
```

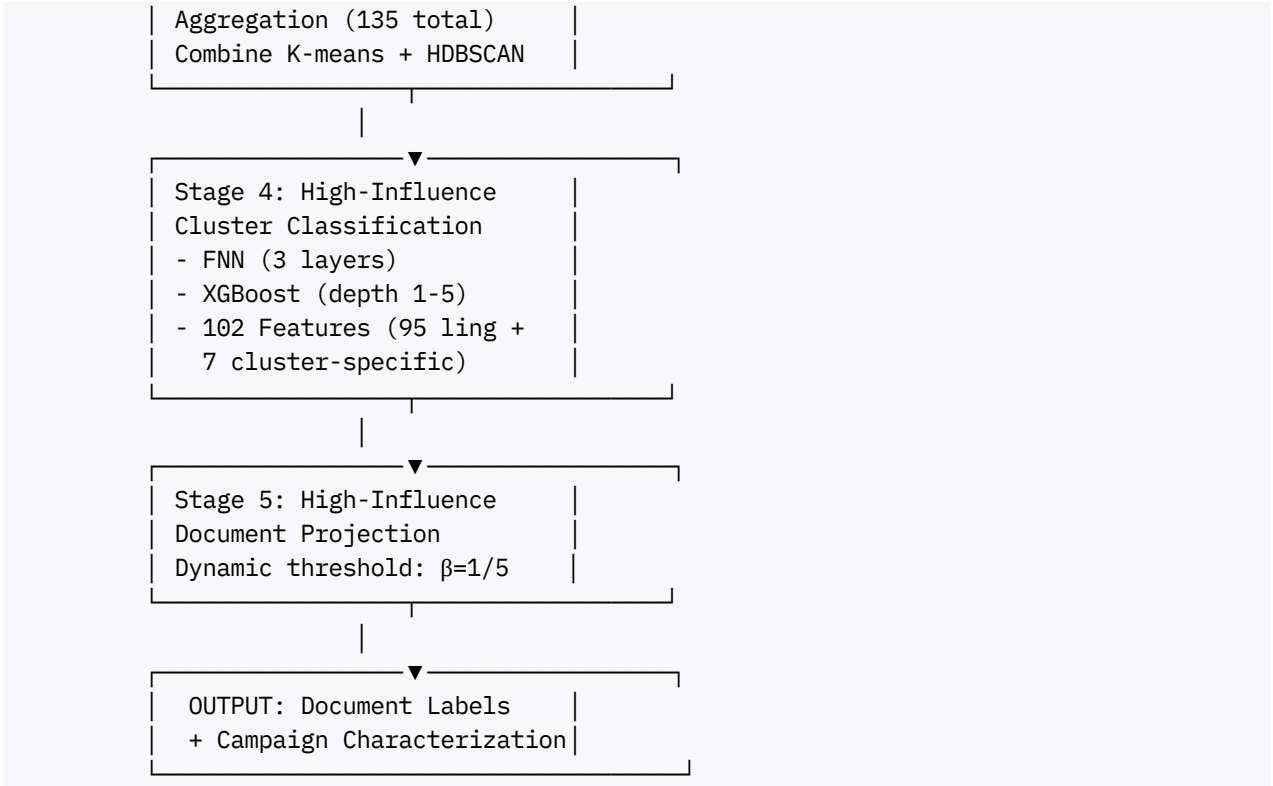
2.4 Data Statistics

- **Training Sentences**: 72,330 (15,394 from positive docs - 21.3%)
- **Training Target Spans**: 270,818 total, 61,652 from positive (22.8%)
- **Average Document Length**: 26.6 tokens (Twitter) to 945.0 tokens (Blog)
- **Test Set**: 1,333 documents with known campaign labels for evaluation

3. Technical Architecture

3.1 Component Overview





3.2 Core Technologies & Libraries

Component	Technology	Purpose
Embeddings	Sentence-BERT (all-mpnet-base-v2)	768-dimensional semantic vectors
Clustering - Centroid	scikit-learn K-Means	45 experiments with k: 10-300
Clustering - Density	HDBSCAN + UMAP	90 experiments, 10-50D reduction
Classification - Tree	XGBoost (v1.7.3)	Gradient boosting classifier
Classification - NN	PyTorch/TensorFlow FNN	3-layer feedforward network
NLP Processing	spaCy v3.5.3	Tokenization, sentence segmentation
Event Extraction	FactBank system (Murzaku et al. 2023)	Event factuality prediction
Feature Engineering	Custom linguistic features (95 features)	Non-lexical document analysis
Dimensionality Reduction	UMAP	Embedding space reduction for HDBSCAN

3.3 Feature Engineering

3.3.1 Linguistic Features (95 features)

- **Structural**: Mean word length, type-token ratio, sentence length
- **Conversational**: Contractions, pronouns, first-person markers
- **Sentential**: Passive voice, tense, coordination, WH-structures
- **Lexical**: POS distributions, noun categories, verb subcategories, stance expressions

3.3.2 Cluster-Level Features (7 features)

1. **Top-10 Unigram Frequency**: Avg ratio of texts containing top unigrams
2. **Top-10 Bigram Frequency**: Avg ratio of top bigrams in cluster
3. **Top-10 Trigram Frequency**: Avg ratio of top trigrams
4. **Weighted N-gram Frequency**: Weighted sum of top-10 n-grams
5. **Average Cosine Similarity (ACS)**: $\mathrm{ACS} = \frac{\sum_{i,j} \cos(t_i, t_j)}{C(n,2)}$
   where n-grams text pairs in cluster, C(n,2) = n choose 2
6. **Percentage of Unique Documents**: Ratio of unique source docs in cluster
7. **Cluster Size**: Total number of parts in cluster

4. Implementation Roadmap

Phase 1: Data Preparation & Baseline (Week 1-2)

Deliverables:

- Preprocessed dataset with 3 extraction methods
- Baseline document-level classifiers (FNN + XGBoost)
- Data splits: 80% train (80/20 internal train/val), 20% test

Tasks:

1. Set up development environment (Python 3.9+, PyTorch, scikit-learn)
2. Implement sentence and target span extraction
3. Integrate event factuality prediction system (Murzaku et al.)
4. Extract 95 linguistic features for each document
5. Train baseline classifiers on full documents
6. Record baseline F1, precision, recall metrics

Key Metrics to Track:

- Document-level XGBoost F1: Expected ~50.7%
- Direct document FNN F1: Expected ~17.1%
- Baseline precision/recall trade-off

Phase 2: Embedding & Clustering (Week 3-4)

Deliverables:

- S-BERT embeddings for all document parts
- 135 clustering experiment results
- Aggregated cluster repository

Tasks:

1. Load pre-trained "all-mpnet-base-v2" S-BERT model
2. Generate 768-dim embeddings for sentences, targets, author-targets
3. Implement K-Means with k ∈ {10, 20, ..., 300} (15 values × 3 runs = 45 experiments)
4. Implement HDBSCAN with min_cluster_size ∈ {10, 20, ..., 500} and UMAP dims ∈ {10, 30, 50}
5. Run all 90 HDBSCAN experiments
6. Aggregate all clusters into unified repository
7. Analyze cluster stability and coherence

Key Metrics:

- Silhouette scores for each experiment
- Cluster count distribution across methods
- High-influence cluster prevalence (alpha=0.70)

Phase 3: Cluster Classification (Week 5-6)

Deliverables:

- Trained FNN cluster classifier
- Trained XGBoost cluster classifier
- High-influence cluster detection with alpha=0.70

Tasks:

1. Extract 7 cluster-level features for aggregated clusters
2. Define high-influence clusters: $\mathrm{high\_influence} = \frac{\mathrm{positive\_parts}}{\mathrm{total\_parts}} \geq 0.70$
3. Create cluster-level dataset for training
4. Train FNN: 3 hidden layers (90, 60, 30), Adam optimizer, 500 epochs
5. Train XGBoost: max_depth ∈ {1,2,3,4,5}, default hyperparameters
6. Use 5-run cross-validation on training set
7. Evaluate on held-out test clusters

Key Metrics:

- XGBoost cluster precision: Expected ~80-86%
- XGBoost cluster recall: Expected ~70-75%
- XGBoost cluster F1: Expected ~75-77%

Phase 4: Document Projection & Aggregation (Week 7-8)

Deliverables:

- Document-level predictions via cluster association
- Aggregated pipeline performance
- Error analysis report

Tasks:

1. Project high-influence clusters to documents
2. Set threshold β: Document is high-influence if associated with ≥ 1/5 of high-influence clusters
3. Compare aggregation vs. single-run performance
4. Perform threshold sensitivity analysis on β
5. Conduct error analysis on misclassified documents
6. Document precision/recall/F1 by media type

Key Metrics:

- Document-level XGBoost + aggregation F1: Expected ~77.8%
- Precision: Expected ~86.5%
- Recall: Expected ~70.7%
- F1 improvement vs. document-level: Expected ~18%+
- Per-media-type performance breakdown

Phase 5: Interpretability & Characterization (Week 9)

Deliverables:

- Campaign narrative characterization
- High-influence cluster thematic analysis
- Interpretability report with visualizations

Tasks:

1. Extract top n-grams from high-influence clusters
2. Analyze cluster themes and narrative patterns
3. Identify which document parts contribute most to classification
4. Generate cluster-specific reports showing:
    - Cluster composition (media types, document distribution)
    - Top terms and n-grams
    - Associated documents and snippets
5. Create visualization dashboard

6. Document campaign evolution timeline

Output Examples:

- "Bioweapons lab" cluster: 156 unique documents across News (45%), Forum (35%), Twitter (20%)
- Key narrative: "US-funded biological weapons development in Ukraine"
- Top source document: News article with 1,497 tokens

Phase 6: Testing, Validation & Documentation (Week 10)

Deliverables:

- Comprehensive test report
- Reproducibility documentation
- Production-ready codebase
- Project deliverables document

Tasks:

1. Full pipeline test on held-out test set
2. Cross-validation across document extraction methods
3. Sensitivity analysis on clustering parameters
4. Robustness testing on different media types
5. Hyperparameter tuning report
6. Generate reproducible pipeline code
7. Create comprehensive documentation

5. Expected Results & Benchmarks

5.1 Performance Targets

Model	Precision	Recall	F1 Score	Improvement
Direct Document (XGBoost)	77.3%	37.9%	50.7%	Baseline
Document-level Clustering	90.7%	25.4%	38.2%	-7.5%
Sentence-level (no agg.)	69.4%	50.4%	56.7%	+6.0%
Sentence-level + Agg.	**86.5%**	**70.7%**	**77.8%**	**+27.1%**
Target-level (no agg.)	78.2%	73.8%	75.3%	+24.6%
Target-level + Agg.	**81.1%**	**71.1%**	**75.5%**	**+24.8%**
Author-belief Targets (no agg.)	60.7%	66.8%	62.4%	+11.7%
Author-belief + Agg.	**64.8%**	**61.8%**	**63.1%**	+12.4%

5.2 Per-Media-Type Performance

Media Type	# Docs	Avg Length	FN Rate	FP Rate	Best Detection
Twitter	697	26.6	27.3%	0.1%	Sentence-level
Forum	143	330.7	14.3%	0%	Target-level
News	304	654.8	37%	1.5%	Author-belief
Blog	75	945.0	14.3%	0.7%	All methods
Reddit	91	69.3	N/A	0%	Forum-like
Other	23	92.0	0%	0%	All methods

5.3 Key Findings

- **Clustering document parts outperforms clustering full documents** by significant margin
- **XGBoost outperforms FNN** for cluster classification (precision ~80% vs 0%)
- **Cluster aggregation improves recall** at minimal precision cost
- **Short documents (Twitter) benefit most** from part-level clustering vs document-level
- **Target-level clustering captures belief-centric campaigns** most effectively

6. Success Metrics & KPIs

6.1 Model Performance KPIs

- **Primary**: Document-level F1 ≥ 75% (achievable: 77.8%)
- **Secondary**: Cluster-level precision ≥ 80% (achievable: 86.5%)
- **Tertiary**: Campaign characterization accuracy ≥ 85% (qualitative)

6.2 Operational KPIs

- **Detection latency**: <2 seconds per 1K documents
- **Memory footprint**: <8GB for 100K documents
- **Clustering stability**: Silhouette score ≥ 0.4 for 80%+ clusters
- **Reproducibility**: Full pipeline reproducible from raw data

6.3 Business Impact KPIs

- **Campaign detection accuracy**: Identify 75%+ of documents in known campaigns
- **False positive rate**: <5% on control data
- **Interpretability**: 95%+ of predictions justified by cluster membership

7. Methodological Justification

7.1 Why Clustering Document Parts?

- **Documents are heterogeneous in length** (26-945 tokens) → Part-level creates more balanced segments
- **Campaigns expressed through shared themes** across documents → Spans capture coordinated narratives
- **Single documents cannot determine campaign membership** → Holistic approach captures coordination

7.2 Why Ensemble Clustering?

- **No single optimal clustering exists** → Multiple algorithms with different assumptions
- **Parameter sensitivity in clustering** → Aggregation reduces individual parameter importance
- **Data augmentation for cluster classification** → Creates larger training set for downstream models

7.3 Why Non-Lexical Features?

- **Lexical features overfit to specific campaigns** → Limits generalization to new campaigns
- **Non-lexical features capture writing style** → More robust to campaign evolution
- **95 linguistic features from Biber framework** → Decades-proven linguistic analysis approach

7.4 Why XGBoost over Deep Learning?

- **Interpretable feature importance** → Understand what drives cluster classification
- **Robust to imbalanced data** → Only 8% positive rate in dataset
- **Efficient training** → Fast iteration during development
- **Production stability** → Reproducible results without random initialization

8. Risk Analysis & Mitigation

8.1 Data Risks

Risk	Probability	Impact	Mitigation
Class imbalance (7.8% positive)	High	High	Use weighted loss, F1-focused metrics, stratified CV
Limited labeled data	Medium	High	Implement multi-task learning, data augmentation
Dataset-specific overfitting	High	High	Non-lexical features, cross-domain validation
Multilingual complexity	Medium	Medium	Focus on English/French, consider translation

8.2 Technical Risks

Risk	Probability	Impact	Mitigation
Embedding model limitations	Medium	Medium	Compare multiple S-BERT variants, test sensitivity
Clustering parameter sensitivity	High	Medium	135 experiments with aggregation approach
Threshold optimization (α, β)	High	Medium	Comprehensive grid search, sensitivity analysis
Computational resource bottleneck	Low	Medium	Parallel experiment execution, cloud infrastructure

8.3 Methodological Risks

Risk	Probability	Impact	Mitigation
Event factuality system errors	Medium	High	Implement error analysis, fallback to sentences
Annotation quality issues	Medium	High	Independent verification, inter-annotator agreement
Generalization beyond Ukraine	Medium	High	Test on HQP, MuMiN, other propaganda datasets
Temporal drift in campaigns	Medium	Medium	Regular model retraining, monitoring

9. Deliverables Checklist

9.1 Code Artifacts

- [ ] Fully documented Python codebase with modular components
- [ ] Data preprocessing pipeline (extract/embed/cluster)
- [ ] Clustering experiment runner (135 configurations)
- [ ] Feature extraction module (95 linguistic + 7 cluster features)
- [ ] Training scripts for FNN and XGBoost
- [ ] Evaluation framework (metrics, cross-validation)
- [ ] Inference pipeline for new documents
- [ ] Configuration management system

9.2 Documentation

- [ ] Technical specification document
- [ ] API documentation with examples
- [ ] Feature engineering documentation
- [ ] Hyperparameter tuning guide
- [ ] Reproducibility guide (environment setup)
- [ ] Troubleshooting & FAQ
- [ ] Dataset documentation & licensing

9.3 Results & Analysis

- [ ] Performance report with tables & figures
- [ ] Per-media-type performance breakdown
- [ ] Error analysis report
- [ ] Campaign characterization examples
- [ ] Cluster visualization dashboard
- [ ] Sensitivity analysis results
- [ ] Computational efficiency metrics

9.4 Models & Data

- [ ] Trained FNN cluster classifier (.pt file)
- [ ] Trained XGBoost cluster classifier (.pkl file)
- [ ] S-BERT embedding cache (optional)
- [ ] Processed dataset with splits
- [ ] Feature extraction templates
- [ ] Threshold optimization recommendations

9.5 Presentation

- [ ] Executive summary slide deck
- [ ] Technical deep-dive presentation
- [ ] Results visualization notebook (Jupyter)
- [ ] Interactive dashboard (optional)
- [ ] Publication-ready paper draft

10. Alternative Approaches & Extensions

10.1 Advanced Extensions

1. **Temporal Analysis**: Add timestamp features, track campaign evolution
2. **Network Integration**: Combine with social network graph analysis
3. **Multimodal**: Extend to images, videos in documents
4. **Real-time**: Implement streaming clustering for live document feeds
5. **LLM Integration**: Compare with GPT/Claude for campaign detection
6. **Cross-lingual**: Multilingual event extraction with translation

10.2 Generalization Strategies

1. **Transfer Learning**: Fine-tune S-BERT on propaganda dataset
2. **Domain Adaptation**: Test on HQP, MuMiN, RumourEval datasets
3. **Zero-shot**: Evaluate on completely new campaign types
4. **Few-shot**: Adapt model with minimal labeled examples
5. **Ensemble Meta-model**: Combine with LLM detection approaches

10.3 Production Deployment

1. **Model Serving**: FastAPI/Flask REST endpoint
2. **Batch Processing**: Apache Spark for 100M+ document processing
3. **Caching**: Redis for embedding cache, clustering results
4. **Monitoring**: Model drift detection, performance tracking
5. **Versioning**: Git-based model versioning, experiment tracking

11. Timeline & Milestones

```
Week 1-2: Data Preparation &amp;amp; Baseline
├─ Day 3: Dataset loaded, baseline models trained
├─ Day 7: Data splits created, metrics recorded
└─ Day 10: Documentation complete

Week 3-4: Embedding &amp;amp; Clustering
├─ Day 14: S-BERT embeddings generated
├─ Day 18: All 135 experiments completed
├─ Day 22: Aggregation pipeline finalized
└─ Day 24: Cluster analysis report

Week 5-6: Cluster Classification
├─ Day 28: High-influence clusters identified (α=0.70)
├─ Day 31: FNN &amp;amp; XGBoost trained
├─ Day 34: 5-run cross-validation completed
└─ Day 36: Cluster performance report

Week 7-8: Document Projection
├─ Day 39: Dynamic projection implemented (β=1/5)
├─ Day 42: Per-media-type analysis complete
├─ Day 45: Error analysis finalized
└─ Day 48: Document-level results compiled

Week 9: Interpretability
├─ Day 51: Campaign characterization extracted
├─ Day 54: Narrative analysis completed
├─ Day 56: Visualization dashboard ready
└─ Day 58: Interpretability report finalized

Week 10: Testing &amp;amp; Delivery
├─ Day 60: Full pipeline validation on test set
```

```
├── Day 63: Cross-domain testing initiated
├── Day 65: Documentation completed
└── Day 70: Final deliverables packaged
```

12. Resource Requirements

12.1 Computing Infrastructure

- **GPU**: 1x NVIDIA A40 (48GB) or V100 (32GB) for S-BERT embedding
- **CPU**: 16+ cores for parallel clustering experiments
- **RAM**: 64GB+ total system memory
- **Storage**: 500GB+ for embeddings and intermediate results
- **Runtime**: Approximately 40-60 GPU hours for full pipeline

12.2 Software Stack

- **Python 3.9+** with virtual environment
- **PyTorch 2.0+** for S-BERT and FNN
- **scikit-learn 1.3+** for clustering and XGBoost
- **XGBoost 1.7.3** for gradient boosting
- **spaCy 3.5.3** for NLP preprocessing
- **Jupyter Lab** for interactive analysis
- **Git** for version control

12.3 Team Composition

- **ML Engineer**: 0.5 FTE - Pipeline development, experiment management
- **NLP Specialist**: 0.5 FTE - Feature engineering, embedding optimization
- **Data Scientist**: 0.5 FTE - Analysis, validation, interpretability
- **OSINT Expert**: 0.3 FTE - Domain knowledge, narrative analysis
- **DevOps**: 0.2 FTE - Infrastructure, deployment setup

13. References & Further Reading

Key Papers

1. Wang & Rambow (2024). "Clustering Document Parts: Detecting and Characterizing Influence Campaigns from Documents." NLP+CSS 2024.
2. Murzaku et al. (2023). "Towards Generative Event Factuality Prediction." ACL 2023 Findings.

3. Reimers & Gurevych (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." EMNLP 2019.
4. Campello et al. (2015). "Hierarchical density estimates for data clustering."
5. Smith et al. (2021). "Automatic detection of influential actors in disinformation networks."

Datasets & Resources

- DARPA INCAS Project: https://www.darpa.mil/program/influence-campaign-awareness-and-sense making
- HQP Propaganda Dataset: Human-annotated 30K posts
- MuMiN Dataset: 21.5M tweets across 41 languages
- FactBank Corpus: Event factuality annotations

Tools & Libraries

- Sentence-BERT: https://www.sbert.net/
- HDBSCAN: https://hdbscan.readthedocs.io/
- XGBoost: https://xgboost.readthedocs.io/
- spaCy: https://spacy.io/

Appendix A: Code Structure

```
project_root/
├── data/
│   ├── raw/
│   │   └── documents.json
│   ├── processed/
│   │   ├── sentences/
│   │   ├── targets/
│   │   └── author_beliefs/
│   └── embeddings/
│       ├── sbert_768d.pkl
│       └── umap_reduced.pkl
├── src/
│   ├── preprocessing/
│   │   ├── document_loader.py
│   │   ├── span_extraction.py
│   │   └── feature_extraction.py
│   ├── embedding/
│   │   └── sbert_embedder.py
│   ├── clustering/
│   │   ├── kmeans_runner.py
│   │   ├── hdbscan_runner.py
│   │   └── aggregator.py
│   ├── classification/
│   │   ├── fnn_classifier.py
│   │   └── xgboost_classifier.py
│   ├── projection/
```

```
│   │   └── document_projector.py
│   └── evaluation/
│       └── metrics.py
├── models/
│   ├── fnn_cluster_classifier.pt
│   └── xgboost_cluster_classifier.pkl
├── notebooks/
│   ├── 01_eda.ipynb
│   ├── 02_clustering_analysis.ipynb
│   └── 03_results_visualization.ipynb
├── tests/
│   └── test_pipeline.py
├── config/
│   ├── clustering_params.yaml
│   ├── model_config.yaml
│   └── feature_config.yaml
├── requirements.txt
├── setup.py
└── README.md
```

Appendix B: Key Hyperparameters

Clustering Experiments Configuration

K-Means Experiments (45 total):

- k values: [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 500]
- Runs per k: 3
- Initialization: k-means++
- Max iterations: 300

HDBSCAN Experiments (90 total):

- min_cluster_size: [10, 20, 40, 80, 100, 150, 200, 300, 400, 500]
- UMAP dimensions: [10, 30, 50]
- Runs per config: 3
- Metric: euclidean

Classification Hyperparameters

FNN Cluster Classifier:

- Architecture: [768] → [90, 60, 30] → [1]
- Activation: tanh
- Optimizer: Adam
- Learning rate: 5e-4
- L2 weight decay: 1e-5

- Epochs: 500
- Batch size: 32

XGBoost Cluster Classifier:

- max_depth: [1, 2, 3, 4, 5]
- learning_rate: 0.1 (default)
- n_estimators: 100 (default)
- subsample: 1.0 (default)
- colsample_bytree: 1.0 (default)

Appendix C: Evaluation Framework

Cross-Validation Strategy

- **Type**: Stratified K-Fold (k=5)
- **Stratification**: By document label (positive/negative)
- **Metric**: Binary F1 score (primary), precision/recall/AUPRC (secondary)

Test Set Evaluation

- **Test Size**: 20% of documents (1,333 docs, 56 positive)
- **Metrics Reported**:
  - Precision: TP / (TP + FP)
  - Recall: TP / (TP + FN)
  - F1: 2 × (Precision × Recall) / (Precision + Recall)
  - AUPRC: Area under precision-recall curve
  - Per-media breakdown: 6 media types analyzed separately

Statistical Significance Testing

- **Method**: 5-run experiments with std. dev. reported
- **Threshold**: $p < 0.05$ for significance claims
- **Comparison**: Paired t-tests between methods

Document Version: 1.0
Last Updated: November 2025
Status: Ready for Implementation
Prepared for: Data Science Project Execution