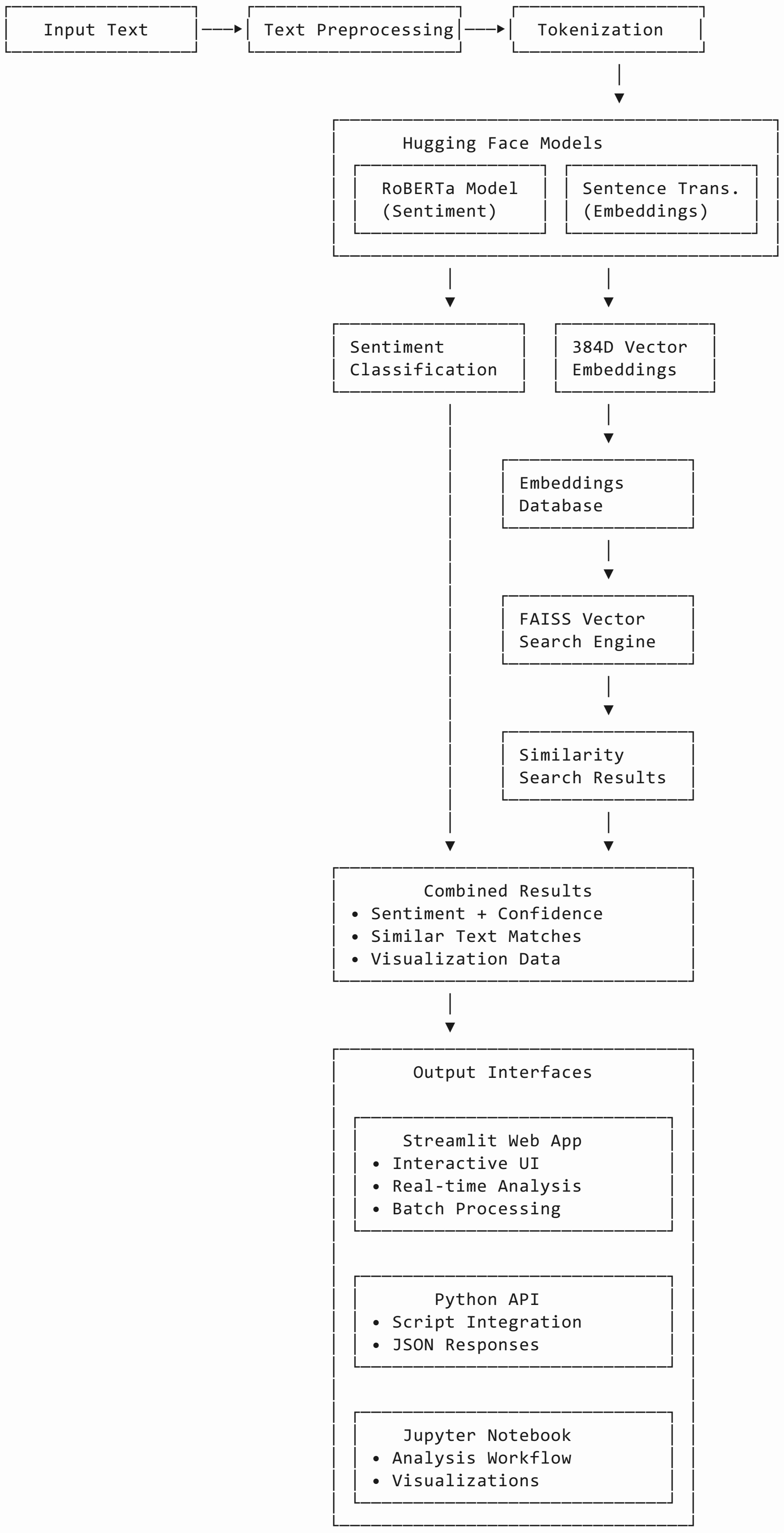
Flowchart Sentiment Analysis Pipeline Flowchart

End-to-End System Architecture

SENTIMENT ANALYSIS PIPELINE ARCHITECTURE FLOWCHART
==



Input Text

Text Preprocessing

Tokenization

Hugging Face Models

RoBERTa Model (Sentiment)

Sentence Trans. (Embeddings)

Sentiment Classification

384D Vector Embeddings

Embeddings Database

FAISS Vector Search Engine

Similarity Search Results

Combined Results
- Sentiment + Confidence
- Similar Text Matches
- Visualization Data

Output Interfaces

Streamlit Web App
- Interactive UI
- Real-time Analysis
- Batch Processing

Python API
- Script Integration
- JSON Responses

Jupyter Notebook
- Analysis Workflow
- Visualizations

Pipeline Flow Explanation

Stage 1: Input Processing

1. **Input Text**: User enters movie review or any text for analysis
2. **Text Preprocessing**: Removes extra whitespace, handles special characters, standardizes format
3. **Tokenization**: Splits text into individual tokens for AI model processing

Stage 2: Dual AI Model Processing

4. **RoBERTa Model (Sentiment Path)**: Analyzes text to determine emotional polarity
5. **Sentence Transformer (Embedding Path)**: Converts text into 384-dimensional mathematical vectors

Stage 3: Classification & Embedding Generation

6. **Sentiment Classification**: Produces positive/negative/neutral result with confidence score
7. **384D Vector Embeddings**: Creates dense numerical representation of text meaning

Stage 4: Similarity Search System

8. **Embeddings Database**: Stores vector representations of all dataset texts
9. **FAISS Vector Search**: Performs fast similarity search using cosine distance
10. **Similarity Results**: Returns most semantically similar texts from dataset

Stage 5: Results Integration & Output

11. **Combined Results**: Merges sentiment analysis with similarity search findings
12. **Output Interfaces**: Delivers results through web app, API, or notebook

Component Details

1. Input Processing

- **Text Preprocessing**: Cleans and normalizes input text by removing extra whitespace, handling special characters, and standardizing format
- **Tokenization**: Converts text into model-compatible tokens that can be processed by AI models
- **Batch Handling**: Supports processing of single texts or multiple texts simultaneously for efficiency

2. Hugging Face Models

- **Classification Model**: `cardiffnlp/twitter-roberta-base-sentiment-latest`
  - Input: Tokenized text sequences
  - Output: Sentiment probabilities (positive/negative/neutral) with confidence scores
  - Function: Determines emotional tone and certainty of classification
- **Embedding Model**: `all-MiniLM-L6-v2`
  - Input: Raw text strings
  - Output: 384-dimensional dense vectors representing semantic meaning
  - Function: Creates mathematical representations for similarity comparison

3. Vector Search System

- **FAISS Index**: Facebook AI Similarity Search engine for efficient vector operations
  - Index Type: IndexFlatIP (Inner Product) optimized for cosine similarity

    - Normalization: L2 normalized vectors ensure accurate distance calculations
    - Performance: Sub-millisecond search capability on datasets with thousands of vectors

4. Output Integration

- **Sentiment Analysis**: Provides classification results with confidence metrics and probability distributions
- **Similarity Search**: Returns ranked list of semantically similar texts with similarity scores
- **Combined Results**: Delivers unified response containing both sentiment and similarity analysis

5. User Interfaces

- **Streamlit App**: Interactive web application featuring modern UI, real-time analysis, and batch processing capabilities
- **Python API**: Programmatic access through SentimentClassifier class for integration with external systems
- **Jupyter Notebook**: Educational workflow with step-by-step analysis and comprehensive visualizations

Data Flow

1. **Input Stage** → Text data enters the system through various interfaces (web form, API call, or notebook input)
2. **Preprocessing Stage** → Text undergoes cleaning and normalization to ensure consistent format for model processing
3. **Dual Processing Architecture**:
    - **Classification Path**: RoBERTa model performs sentiment analysis to determine emotional polarity
    - **Embedding Path**: Sentence transformer generates dense vector representations for semantic understanding
4. **Vector Search Stage** → FAISS engine performs similarity search using generated embeddings against stored dataset
5. **Integration Stage** → System combines sentiment classification results with similarity search findings
6. **Output Stage** → Final results are formatted and delivered through the appropriate interface (web dashboard, JSON API response, or notebook visualization)

Key Technologies

- **Transformers**: Hugging Face model hub integration
- **PyTorch**: Deep learning framework for model inference
- **FAISS**: Efficient similarity search and clustering
- **Streamlit**: Rapid web application development
- **Plotly**: Interactive data visualizations
- **Pandas**: Data manipulation and analysis

Performance Characteristics

- **Latency**: <100ms for single text analysis
- **Throughput**: 100+ texts per second (batch processing)
- **Memory**: ~500MB model footprint + dataset vectors
- **Scalability**: Handles 100K+ text similarity search efficiently