Text Analytics

Ramaseshan Ramachandran

January 25, 2025

WHAT IS TEXT ANALYTICS?



- Converts unstructured text into insights
- Extracts patterns and trends
- Uses NLP, machine learning, and statistics
- Analyzes data from diverse sources

IMPORTANCE OF TEXT ANALYTICS



- Unstructured text data is vast
- ► Text data is messy and variable
- Makes data measurable and valuable

BENEFITS OF TEXT ANALYTICS



- Understand customers and societal trends
- ► Enhance decision-making and planning
- Drive engagement and innovation

INFORMATION RETRIEVAL (IR)



- Searches documents and metadata
- Extracts relevant data to queries
- Applies to large text collections

- Boolean Retrieval Model
- Vector Space Model (TF-IDF)
- Probabilistic Retrieval Model
- Latent Semantic Indexing (LSI)
- ▶ BM25 Algorithm

TEXT CLASSIFICATION



- Categorizes text into predefined labels
- Applications: spam detection, sentiment analysis
- Automates document organization

- Naive Bayes Classifier
- Support Vector Machines (SVM)
- Logistic Regression
- Decision Trees and Random Forests
- Deep Learning (CNNs, RNNs, Transformers)

CLUSTERING



- Groups similar texts together
- No predefined labels required
- Useful for exploratory analysis

- K-Means Clustering
- Hierarchical Clustering
- ► DBSCAN (Density-Based Clustering)
- Gaussian Mixture Models (GMMs)
- Spectral Clustering

SENTIMENT ANALYSIS



- Determines text's emotional tone
- Classifies as positive, negative, or neutral
- Applications: marketing, feedback analysis

- Lexicon-Based Approaches
- Rule-Based Sentiment Analysis
- Machine Learning-Based Approaches
- Neural Networks (LSTMs, GRUs, BERT)
- Pretrained Models (RoBERTa, GPT)

ENTITY RECOGNITION



- Identifies entities in text (e.g., names)
- Categorizes into people, places, etc.
- Useful for automated content analysis

- ► Hidden Markov Models (HMMs)
- Conditional Random Fields (CRFs)
- Maximum Entropy Models
- ▶ Neural Networks (BiLSTM + CRF)
- Pretrained Models (SpaCy, Hugging Face)

TOPIC MODELING



- Discovers themes in document collections
- Applications: summarization, recommendations

Algorithms

- ► Latent Dirichlet Allocation (LDA)
- Non-Negative Matrix Factorization (NMF)
- ► Latent Semantic Analysis (LSA)
- Gibbs Sampling for LDA
- Neural Topic Models (ProdLDA, BERTopic)

OTHER ADVANCED TECHNIQUES



- ▶ Word Embedding Models (Word2Vec, GloVe, FastText)
- Sentence Embeddings (Sentence-BERT)
- Attention Mechanisms
- Transformers (BERT, GPT, T5)
- Text Summarization (Extractive and Abstractive)

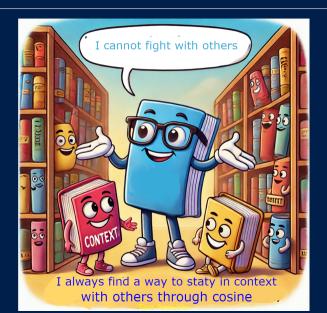
CONCLUSION



- ► Text analytics unlocks data potential
- Drives decisions and innovation
- Benefits industries like finance and healthcare
- A variety of algorithms drive text analytics
- ► Techniques range from statistical to neural
- ► Tailor solutions based on use case

WORD EMBEDDING CANNOT FIGHT WITH OTHERS. WHY





WHAT IS SCIKIT-LEARN?



- Definition: Scikit-learn is a Python library for machine learning, providing tools for:
 - Classification
 - Regression
 - Clustering
 - Dimensionality reduction
 - Preprocessing and more
- Key Features:
 - ▶ Built on NumPy, SciPy, and matplotlib
 - Simple and efficient tools for predictive data analysis
 - Open source

WHY USE SCIKIT-LEARN IN TEXT ANALYTICS?



► Text Analytics Focus:

- Natural Language Processing (NLP) tasks
- Feature extraction (e.g., bag-of-words, TF-IDF)
- Building predictive models

Advantages:

- Wide range of algorithms (SVMs, Naive Bayes, etc.)
- User-friendly API for rapid prototyping
- Extensive documentation and community support

COMMON USE CASES IN TEXT ANALYTICS



Examples:

- Spam detection
- Sentiment analysis
- Topic modeling
- Document classification

▶ Techniques:

- Preprocessing: Tokenization, stemming, lemmatization
- Vectorization: TF-IDF or CountVectorizer
- ▶ Model training: Logistic regression, SVMs, etc.

TYPICAL WORKELOW IN SCIKIT-LEARN



1. Data Preprocessing:

- Cleaning text data (e.g., removing stop words)
- Vectorization (e.g., TF-IDF)
- 2. Model Selection:
 - Choosing an algorithm (e.g., Naive Bayes)
- 3. Model Training:
 - model.fit(X_train, y_train)
- 4. Model Evaluation:
 - Metrics like accuracy, precision, recall
- 5. Prediction:
 - model.predict(X test)

EXAMPLE: TEXT CLASSIFICATION WORKFLOW



Dataset: Sentiment Analysis on Product Reviews

- 1. Load dataset
- 2. Preprocess text (lowercase, remove punctuation, etc.)
- 3. Vectorize with TF-IDF
- 4. Train model (e.g., Logistic Regression)
- 5. Evaluate using accuracy and F1-score

CODE EXAMPLE:



```
from sklearn.feature extraction.text import TfidfVectorizer
from sklearn.model selection import train test split
from sklearn.linear model import LogisticRegression
from sklearn.metrics import accuracy score
vectorizer = TfidfVectorizer()
X = vectorizer.fit transform(text data)
X_train, X_test, y_train, y_test = train_test_split(X, labels, te
model = LogisticRegression()
model.fit(X train, y train)
predictions = model.predict(X test)
print("Accuracy:", accuracy score(y test, predictions))
```

STRENGTHS AND LIMITATIONS



Strengths:

- Easy to use and integrate
- Extensive support for text-related tasks
- Scalability for moderate-sized datasets

Limitations:

- Not designed for deep learning
- ► Limited support for out-of-core learning

RESOURCES TO LEARN MORE



- Official Documentation: https://scikit-learn.org
- Tutorials:
 - "Getting Started with Scikit-Learn" (Blog/Video)
 - Kaggle courses on ML
- Recommended Books:
 - Python Machine Learning by Sebastian Raschka
 - Introduction to Machine Learning with Python by Andreas Müller



Questions?