Exam Report

DSK 822: News and Market Sentiment Analytics

Szimona Szternak

19 Dec 2025

1 Introduction

This project demonstrates the application of Natural Language Processing (NLP) techniques to extract insights from a corpus of Reuters news articles. The goal was to showcase skills in text analysis, classification, retrieval, and sentiment evaluation, using the tools and methods covered in the course.

2 Dataset

The dataset used in this project is a modified version of the Reuters-21578 corpus, available on Kaggle [5].

It contains multiple CSV files representing different splits (e.g., ModApte, ModHayes, ModLewis) of the original corpus. For each article, the dataset provides the following fields:

- *text:* Full text of the news article.
- *title:* Article title.
- *topics:* List of topics assigned to the article.
- *places, people, orgs, exchanges:* Named entity information.
- *date:* Publication date.
- *text_type, lewis_split, cgis_split, old_id, new_id:* Additional metadata for corpus management.

For this project, the ModApte split was primarily used. A binary label was created for classification tasks: articles containing the topic "earn" were labeled as 1 (earnings-related), and all others as 0. This allowed for supervised training of LSTM and DistilBERT models and served as a reference for zero-shot and retrieval-augmented analyses. The dataset also provides a rich source for unsupervised analyses, such as word frequency distributions, Zipf curves, and sentiment scoring.

3 NLP Tools and Motivation

This project leverages a combination of classical and modern NLP tools to extract meaningful insights from the Reuters news corpus. The chosen tools reflect both foundational methods and state-of-the-art approaches suitable for financial news analysis.

3.1 Text Preprocessing and Analysis

- **Regular expressions and tokenization:** Used for cleaning and splitting text into words, enabling analysis such as word frequency counts and Zipf curves.
- **Counter from Python collections:** Efficiently computes word frequencies to identify patterns in text distributions.
- **Motivation:** Preprocessing is essential to remove noise and prepare textual data for subsequent modeling. Zipf analysis provides an exploratory understanding of language patterns in the corpus.

3.2 Neural Network Classifiers

- **LSTM (Long Short-Term Memory):** Sequential model with embedding and recurrent layers, trained to classify articles based on binary labels ("earn" vs. others). Captures temporal dependencies in text sequences, suitable for short- to medium-length financial news articles.
- **DistilBERT (Transformer-based model):** Fine-tuned pre-trained transformer model leveraging contextual embeddings for improved semantic understanding. Particularly effective for capturing nuanced meaning in financial texts where word order and context matter. [4]
- **Motivation:** LSTM serves as a classical deep learning baseline, while DistilBERT demonstrates the advantage of transformer-based contextual understanding in classification tasks.

3.3 Embedding-Based Zero-Shot Classification

- **Sentence-BERT (all-MiniLM-L6-v2):** Generates dense vector embeddings for both texts and candidate labels. Classifies articles without supervised training by computing cosine similarity between text and label embeddings. [2]
- **Motivation:** Zero-shot embeddings allow flexibility in labeling and testing new topics, particularly when labeled data is limited, reducing the dependency on annotated datasets.

3.4 Sentiment Analysis

- **VADER (Valence Aware Dictionary for Sentiment Reasoning):** Rule-based sentiment analyzer tuned for social and financial text. Provides compound scores from -1 (negative) to +1 (positive). [3]
- **Motivation:** Financial sentiment is a key signal in market analysis; VADER is lightweight, interpretable, and effective for short financial news articles.

3.5 Retrieval-Augmented Generation (RAG)

- **Sentence embeddings + generative models (BART large):** Retrieves top-k semantically relevant articles to a user query and generates natural language responses.
- **Motivation:** Enables knowledge-based question answering, summarization, and insight generation from large text corpora without requiring task-specific fine-tuning.

4 Results

4.1 Zipf Analysis

The first 10 Reuters articles were analyzed to visualize word frequency distributions. Zipf curves were plotted on a log-log scale, showing a power-law distribution typical of natural language. [1]

- Common function words dominated (e.g., "the", "and"), while most words occurred infrequently.
- This analysis confirms expected textual patterns in the corpus and provides insight into the vocabulary richness of financial news.
- Results saved as *zipf_curves_first_10_articles.png*

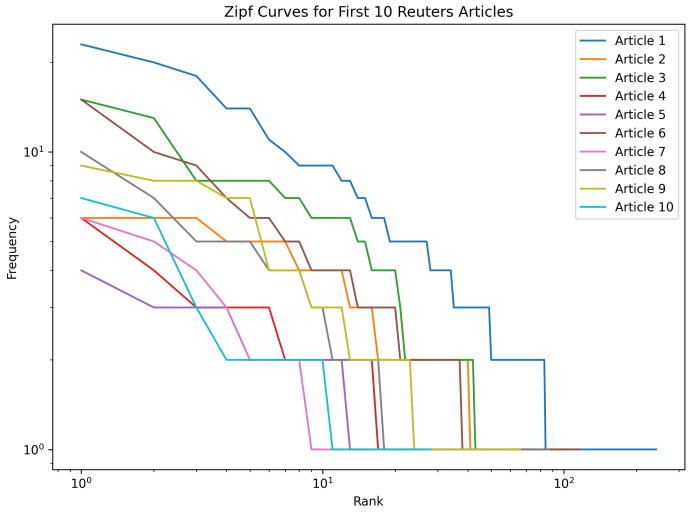

Figure 1: Zipf Curves of the first 10 articles

4.2 Classification Performance

LSTM achieved strong performance (96.1%), effectively capturing sequential patterns in the text.

DistilBERT outperformed the LSTM (98.3%), benefiting from contextual embeddings that capture nuanced semantic information.

Zero-shot embeddings demonstrated moderate performance (83.0%), showing that embedding-based similarity can classify articles without supervised fine-tuning, but with some loss of accuracy compared to task-specific models.

Results saved as *model_comparison_results.csv*

Table 1: Classification Accuracy of Different Models

Model	**Accuracy**
LSTM	0.961
DistilBERT	0.983
Zero-Shot Embeddings	0.830

4.3 Sentiment Analysis

Sentiment scores were computed using **VADER**, producing compound values between -1 (negative) and +1 (positive). Positive scores correlated with favorable financial news, while negative scores reflected adverse events or market reactions.

Sentiment scores were stored alongside article titles and text in CSV files (*train_sentiment_scores.csv* and *test_sentiment_scores.csv*) for further analysis.

4.4 Retrieval-Augmented Generation (RAG)

A RAG pipeline was used to answer queries based on the news corpus:

- **Query example:** "Which articles discuss earnings?"
- **Retrieved articles:** Top-k relevant articles based on semantic similarity.
- **Generated summary:**Synthesized answer combining content from retrieved articles.

This demonstrates how retrieval-augmented generation enables knowledge extraction and query-driven summarization from a large corpus.

5 Documentation

All code and data used in this project are publicly available in the following GitHub repository:

`https://github.com/SzimonaSzternak/news_and_market_szszt24`

The repository contains:

- Python script for data preprocessing, Zipf analysis, LSTM and DistilBERT classification, zero-shot embedding classification, RAG pipeline, and sentiment analysis.
- Resulting CSV outputs, and plots.
- Datasets

This ensures full reproducibility of the results presented in this report and allows others to explore, extend, or adapt the analyses.

References

[1] Weisberg, Sanford. *Applied Linear Regression, 4th Edition*. Wiley, 2014.

[2] Reimers, Nils and Gurevych, Iryna. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." EMNLP 2019.

[3] Hutto, C.J., and Gilbert, E. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text." Eighth International AAAI Conference on Weblogs and Social Media (ICWSM-14), 2014.

[4] Hugging Face Transformers Documentation. `https://huggingface.co/docs/transformers`, 2025.

[5] Uncovering Financial Insights with the Reuters-2 Dataset. Kaggle. `https://www.kaggle.com/datasets/thedevastator/uncovering-financial-insights-with-the-reuters-2`, 2025.

[6] Vedels, Christian. "News and Market Sentiment Analytics – GitHub repository with course codes and lectures." `https://github.com/christianvedels/News_and_Market_Sentiment_Analytics/tree/main`, 2025.

[7] OpenAI. "ChatGPT (GPT-4/5) - Generative AI assistance for code and report writing." `https://openai.com/`, 2025.

[8] GitHub Copilot. "GitHub Copilot – AI-powered coding assistant integrated into Visual Studio Code." `https://github.com/features/copilot`, 2025.