



Earnings Call Transcript Analyzer

A comprehensive AI-powered tool to analyze earnings call transcripts and generate investment insights using NLP, sentiment analysis, and financial pattern recognition.



Features

- **PDF Text Extraction:** Automatic extraction from earnings call transcripts
- **Financial Metrics Detection:** Identifies revenue, profit, EBITDA, margins, and trends
- **Sentiment Analysis:** Analyzes management tone and Q&A interactions
- **Risk Detection:** Flags potential risk factors and concerns
- **Investment Recommendation:** Generates bullish/neutral/bearish signals
- **Interactive Dashboard:** Beautiful Streamlit interface with visualizations
- **Export Capabilities:** Download analysis results as JSON



Prerequisites

- Python 3.8 or higher
- pip package manager
- 2-3 GB free disk space (for ML models)



Installation

1. Clone or Download the Project



bash

```
# Create project directory
```

```
mkdir earnings-analyzer
```

```
cd earnings-analyzer
```

2. Create Virtual Environment (Recommended)



bash

Create virtual environment

```
python -m venv venv
```

Activate virtual environment

On Windows:

```
venv\Scripts\activate
```

On macOS/Linux:

```
source venv/bin/activate
```

3. Install Dependencies



bash

Install required packages

```
pip install -r requirements.txt
```

Download spaCy English model

```
python -m spacy download en_core_web_sm
```

4. Project Structure

Ensure your project has the following structure:



earnings-analyzer/

├── analyzer.py # Main analysis engine (enhanced version)

├── app.py # Streamlit application

├── requirements.txt # Python dependencies

└── README.md # This file

Usage

Running the Streamlit App



bash

```
streamlit run app.py
```

The application will open in your default web browser at `http://localhost:8501`

Using the App

1. **Upload PDF:** Click the file uploader and select an earnings call transcript (PDF format)
2. **Optional:** Enter the company name in the sidebar
3. **Analyze:** Click the "🚀 Analyze Transcript" button
4. **Review Results:**
 - Investment View (Bullish/Neutral/Bearish)
 - Financial metrics and trends
 - Management commentary highlights
 - Sentiment analysis with gauges
 - Risk factors
5. **Export:** Download analysis as JSON for your records

Programmatic Usage



python

```
from analyzer import analyze_transcript

# Open PDF file
with open('earnings_transcript.pdf', 'rb') as pdf_file:
    results = analyze_transcript(pdf_file)

# Access results
print(results['investment_view']['verdict'])
print(results['financials'])
print(results['sentiment'])
```

Understanding the Results

Investment View Score (0-9 points)

- **Bullish** 🟢 : Score $\geq 6/9$ (65%+)
- **Neutral** 🟡 : Score 4-5/9 (40-65%)
- **Bearish** 🔴 : Score $\leq 3/9$ (<40%)

Scoring Breakdown:

- **Financial Metrics** (2 pts): Revenue, profit, EBITDA disclosure

- **Commentary Balance** (2 pts): Positive vs concerning statements
- **Management Sentiment** (2 pts): Tone and confidence level
- **Q&A Sentiment** (2 pts): Interaction quality and tone
- **Risk Assessment** (1 pt): Number of risk flags identified

Sentiment Scores

- **Positive:** Score > 0.05
- **Neutral:** Score between -0.05 and 0.05
- **Negative:** Score < -0.05

Scores range from -1 (extremely negative) to +1 (extremely positive)

Features Explained

Financial Analysis

- Extracts key metrics using regex pattern matching
- Identifies growth trends (YoY, QoQ)
- Supports multiple currencies (₹, Rs., USD, \$)
- Recognizes various units (Crores, Million, Billion)

Commentary Analysis

- Identifies positive highlights (15 keywords)
- Flags concerns and challenges (14 keywords)
- Generates executive summary using T5 model
- Limits results to top 10 for readability

Sentiment Analysis

- Uses VADER (Valence Aware Dictionary and sEntiment Reasoner)
- Separate analysis for management and Q&A sections
- Visual gauge charts for easy interpretation

Risk Detection

- Scans for 17 risk-related keywords
- Extracts full sentences for context
- Limits to top 15 most relevant risks

Configuration

Customizing Keywords

Edit analyzer.py to modify keyword lists:



python

```
# In CommentaryAnalyzer class
POSITIVE_KEYWORDS = ['strong', 'growth', ...] # Add your keywords
NEGATIVE_KEYWORDS = ['headwinds', 'challenging', ...]

# In RiskDetector class
RISK_KEYWORDS = ['risk', 'debt', ...]
```

Adjusting Financial Patterns

Modify regex patterns in `FinancialAnalyzer.PATTERNS` to match your transcript format.

Changing Score Thresholds

In `InvestmentViewGenerator.generate()`, adjust:



python

```
if score_pct >= 65: # Change threshold for Bullish
    verdict = InvestmentView.BULLISH.value
```

Troubleshooting

Common Issues

1. PDF extraction fails

- Ensure PDF contains extractable text (not scanned images)
- Try converting scanned PDFs with OCR first

2. Models not loading



bash

```
# Reinstall spaCy model
python -m spacy download en_core_web_sm --force
```

3. Out of memory errors

- Reduce PDF size or split large transcripts
- Close other applications
- Increase system RAM allocation

4. Slow performance

- First run loads models (takes 1-2 minutes)
- Subsequent analyses are faster
- Consider using GPU for transformer models

5. Streamlit won't start



bash

```
# Check if port 8501 is in use
streamlit run app.py --server.port 8502
```

Best Practices

1. **Use Official Transcripts:** Download from company investor relations pages
2. **Check PDF Quality:** Ensure text is extractable, not images
3. **Compare Multiple Quarters:** Analyze trends over time
4. **Combine with Other Research:** Use as supplementary tool, not sole investment basis
5. **Verify Financial Numbers:** Cross-check extracted metrics with official reports

Limitations

- **Text Extraction:** May miss content in complex PDF layouts
- **Financial Metrics:** Depends on consistent formatting in transcripts
- **Sentiment Analysis:** May not capture all nuances and context
- **Pattern Matching:** Regex patterns may need customization for different formats
- **Not Financial Advice:** Results are for research purposes only

Technical Stack

- **PDF Processing:** PyMuPDF (fitz)
- **NLP:** spaCy (en_core_web_sm)
- **Sentiment:** VADER Sentiment
- **Summarization:** T5-small (Hugging Face Transformers)
- **UI:** Streamlit
- **Visualization:** Plotly

Dependencies



```
streamlit==1.31.0
pymupdf==1.23.26
spacy==3.7.2
vaderSentiment==3.3.2
transformers==4.36.2
torch==2.1.2
plotly==5.18.0
pandas==2.1.4
```

Version History

v2.0 (Enhanced Version)

- Object-oriented architecture
- Lazy model loading
- Improved error handling
- Better logging
- Type hints throughout
- Enhanced scoring system

v1.0 (Original)

- Basic analysis functionality
- Simple script structure

Contributing

Suggestions for improvements:

1. Add support for more financial metrics
2. Implement multi-language support
3. Add comparison across multiple transcripts
4. Export to PDF/Excel formats
5. Add historical trend analysis

License

This project is for educational and research purposes. Not financial advice.

Support

For issues or questions:

1. Check the Troubleshooting section
2. Review error logs in the terminal
3. Verify all dependencies are installed correctly

Educational Use

This tool demonstrates:

- NLP application in finance
- Sentiment analysis techniques
- Document processing pipelines
- Web application development with Streamlit
- ML model integration

Disclaimer: This tool provides analysis for research purposes only and should not be considered as financial or investment advice. Always conduct thorough research and consult with financial professionals before making investment decisions.