SEC 10-K Risk Factor Intelligence

NLP Portfolio Project Outline

Thesis

"Can we automatically classify corporate risk disclosures into meaningful categories and identify which risks are boilerplate vs. materially substantive?"

This frames NLP as a decision-support tool for analysts, compliance teams, or investors—more practical and differentiated than typical sentiment-to-returns approaches.

Strategic Value: Differentiation

Common Approach	Your Differentiated Approach
Sentiment scoring → predict returns	Risk taxonomy classification
Aggregate tone metrics	Document-level + sentence-level analysis
Backward-looking correlation	Forward-looking: detect new/emerging risks
Single model	Progressive complexity: TF-IDF → BERT

Dataset

EDGAR-CORPUS on Hugging Face

- URL: https://huggingface.co/datasets/eloukas/edgar-corpus
- 10-K filings from 1993–2020, pre-parsed and ready to load
- Focus on **Item 1A: Risk Factors** section (required since 2005)

Alternative: Use edgar-crawler (https://github.com/nlpaueb/edgar-crawler) to pull more recent filings (2021–2024) for current data.

Project Phases

Phase 1: Data Acquisition & Exploration

Tasks:

- Load EDGAR-CORPUS from Hugging Face
- Filter to 10-K filings, 2006–2020 (Item 1A required from 2005)
- Extract Item 1A (Risk Factors) section
- Sample 3–5 filings manually to understand structure
- Basic EDA: document lengths, word counts by year, filing volume by industry (SIC codes)

Deliverable: Clean dataset of ~50,000+ Risk Factor sections with metadata (ticker, year, industry)

Phase 2: Text Preprocessing Pipeline

Tasks:

- Lowercase, remove HTML artifacts
- Handle legal boilerplate markers ("Item 1A", section headers)
- Sentence segmentation (critical—risk factors are paragraph-dense)
- Tokenization choices: word-level vs. subword (for transformers later)
- Optional: Remove tables/numeric content or flag them

Key Decision: Work at both document-level and sentence-level. Sentence-level enables classification of individual risks.

Deliverable: Preprocessing module (reusable Python code)

Phase 3: Risk Taxonomy Development

Tasks:

- Review SEC guidance and academic literature on risk categories
- Define 8–12 risk categories (examples below)
- Manually label 500–1,000 sentences as training data

Risk Category	Risk Category
Regulatory/Legal	Cybersecurity/Technology
Competitive/Market	Macroeconomic
Operational	Supply Chain
Financial/Liquidity	Reputational
Environmental/Climate	Key Personnel

Deliverable: Labeled training set + taxonomy documentation

Phase 4: Baseline Models

Tasks:

- **TF-IDF + Logistic Regression** — explainable baseline
- **TF-IDF + SVM** — often strong for text classification
- **TF-IDF + Random Forest** — for comparison
- Evaluation: accuracy, macro F1, confusion matrix
- Error analysis: which categories are confused?

Why this matters: Interviewers will ask "why did you use BERT?" You need to show you tried simpler methods first.

Deliverable: Baseline results table, confusion matrices

Phase 5: Word Embeddings & Neural Approaches

Tasks:

- **Word2Vec / GloVe embeddings** + LSTM or simple feedforward
- Compare pre-trained embeddings vs. domain-specific
- Optional: Train embeddings on your corpus
- Optional: CNN for text classification

Deliverable: Embedding-based model results, comparison to baselines

Phase 6: Transformer Fine-Tuning

Tasks:

- Fine-tune **DistilBERT** or **RoBERTa** on labeled data
- Use Hugging Face transformers library
- Handle long documents: chunking strategy (10-K risk sections can exceed 5,000 words; BERT max is 512 tokens)
- Hyperparameter tuning: learning rate, batch size, epochs
- Evaluate: F1, precision/recall by category

Stretch goal: Try FinBERT (pre-trained on financial text) and compare to generic BERT

Deliverable: Fine-tuned model, performance comparison table

Phase 7: Boilerplate vs. Material Risk Detection

This is your key differentiator.

Approach Options:

1. **Year-over-year change detection:** Compare company's 2020 vs. 2019 risk factors. Flag sentences that are new or significantly modified.
2. **Similarity scoring:** Compute cosine similarity of a company's risk section to industry peers. High similarity = boilerplate.
3. **Novelty detection:** Use embeddings to identify sentences that are outliers relative to the corpus.

Tasks:

- Implement change detection for a sample of companies
- Visualize: "What new risks did Company X disclose in 2020 vs. 2019?"
- Quantify: % boilerplate vs. material by industry

Deliverable: Boilerplate detection module, case study examples

Phase 8: Business Value & Interpretation

Tasks:

- Frame results for a business audience
- Example: "An analyst reviewing 50 10-Ks could reduce review time by X% using automated classification"
- Example: "Model correctly identified 3 companies that disclosed new cyber risks before breach announcements"
- Calculate potential efficiency gains
- Discuss limitations honestly

Deliverable: Executive summary section for write-up

Phase 9: Deployment Artifact

Options (pick one):

Option A: Streamlit App (Recommended)

- User inputs ticker + year range
- App displays: risk category breakdown, year-over-year changes, boilerplate score
- Interactive visualizations

Option B: Jupyter Notebook Report

- Polished, narrative-driven notebook suitable for sharing
- Clear visualizations, minimal code visible

Option C: FastAPI Endpoint

- POST text → returns risk categories + confidence scores
- Demonstrates production-readiness

Technical Stack

Component	Tool
Data loading	Hugging Face datasets, pandas
Preprocessing	spaCy, NLTK, regex
Traditional ML	scikit-learn
Deep Learning	PyTorch, Hugging Face transformers
Embeddings	Gensim (Word2Vec), Hugging Face
Visualization	matplotlib, seaborn, plotly
Deployment	Streamlit
Version control	Git/GitHub

Skills Demonstrated

Skill	Evidence
NLP preprocessing	Custom pipeline for messy legal text
Feature engineering	TF-IDF, embeddings, chunking strategies
Classical ML	Logistic regression, SVM baselines
Deep learning	LSTM, transformer fine-tuning
Hugging Face ecosystem	Loading datasets, fine-tuning models
Handling long documents	Chunking/aggregation for BERT limits
Model evaluation	F1, confusion matrices, error analysis
Domain expertise	Financial text, SEC filings
Deployment	Interactive Streamlit app
Communication	Business value framing

Final Deliverables

1. **GitHub repository** — clean code, README, requirements.txt

2. **Streamlit app** — deployed on Streamlit Cloud
3. **Technical write-up** — methodology, results, limitations (README or separate PDF)
4. **Portfolio page** — summary for your website

Risks & Mitigations

Risk	Mitigation
Manual labeling is slow	Start with 500 labels; use active learning or semi-supervised if needed
Long documents exceed BERT limits	Chunking + aggregation strategy (mean pooling, hierarchical attention)
Weak classification results	Frame as exploratory; baseline comparison still shows methodology
Scope creep	Strict phase gates; cut boilerplate detection if behind

Key Resources

- **Dataset:** https://huggingface.co/datasets/eloukas/edgar-corpus
- **Data Crawler:** https://github.com/nlpaueb/edgar-crawler
- **SEC EDGAR:** https://www.sec.gov/search-filings/edgar-search-assistance/accessing-edgar-data
- **Loughran-McDonald Dictionary:** Standard financial sentiment lexicon for benchmarking