Form 10-K Parser: NLP techniques to Analyze Financial Documents

Christopher Li, Jaylon McDuffie, Howard Appel, Sam Murshed
New York University
May 2025

Form 10-K Parser: NLP techniques to Analyze Financial Documents

Abstract

Financial documents such as SEC Form 10K filings are very complicated and difficult for investors to read because of their unclear structure and complex language. So we sought out to create a simple NLP parser that automatically pulls important financial information out of these hard to read documents. Our parser uses basic NLP tools like TFIDF and ngrams with more modern techniques including transformers like BERT and automatic labeling using GPT 3.5. We tested the parser on recent financial reports from major companies and got very good results. It was able to consistently find key financial data including earnings, EBITDA, net income, risk factors, and references to artificial intelligence. Our parser got weighted average F1 scores above 80 percent and in some tests even nearly 88 percent. This study provides a free and easy to use NLP tool that reduces the time needed to label documents by hand and makes complicated financial information clearer for investors.

Introduction

Document parsing is a process that turns complicated unstructured text into clear structured information, which helps people quickly find and understand important data. Parsing SEC Form 10K documents is useful because they give investors important details about company performance, future risks, and important projects. But these documents are challenging to read because they use complicated financial terms, have inconsistent formatting, have unclear wording and can potentially be up towards 100+ pages.

The goal of our research is to develop a simple NLP parser specifically designed to handle these difficult financial documents. To do this we used traditional NLP methods such as TFIDF vectorization and ngram analysis together with machine learning methods such as BERT transformers and GPT 3.5. Our parser automatically identifies and extracts key financial information like earnings, EBITDA, net income, operating expenses, risk factors, and artificial intelligence mentions.

Our paper goes in depth about our open source and easy to use NLP parser that extracts important financial data from SEC Form 10K documents. We tested the parser using reports from different companies and found that it works effectively, making financial filings clearer and easier to analyze for investors, researchers and anyone else interested.

Literature Review and Related Work

Natural language processing (NLP) has extensively been applied to financial document analysis, particularly SEC Form 10-K filings, due to their dense informational content relevant to investors. Term Frequency-Inverse Document Frequency (TF-IDF), introduced by Salton and Buckley (1988), is a foundational method for capturing term importance within documents. Their method quantifies the importance of a term/word by combining its frequency within a document and its rarity across the corpus. This in turn creates numerical representations which can be utilized for text classification tasks. While our approach does employ TF-IDF to establish an interpretable baseline, we expanded on this by integrating modern machine learning models that better capture semantic nuances beyond word frequency alone.

Expanding beyond individual words, n-gram features introduced by Shannon (1951) analyze phrases, enhancing contextual understanding. Shannon's approach enables models to better grasp nuanced meanings, such as recognizing the importance of phrases like “net income” or “risk exposure.” Our NLP pipeline incorporates n-grams as a crucial feature in TF-IDF vectorization to enhance classification accuracy. However, unlike Shannon's purely statistical use

of n-grams, we combine these with machine learning classifiers to improve semantic context recognition further.

The advent of Transformer-based architectures significantly advanced NLP capabilities. Bidirectional Encoder Representations from Transformers (BERT), developed by Devlin et al. (2019), employs attention mechanisms allowing models to learn contextually rich representations by considering words in their full sentence context simultaneously from both directions. This method excels at interpreting linguistic nuances, significantly improving classification accuracy compared to traditional methods. Similar to Devlin et al., we leverage BERT's deep contextual understanding; however, we specifically fine-tune our models on financial documents to address the unique complexities inherent in SEC Form 10-K filings.

Large Language Models such as GPT-3 introduced by Brown et al. (2020), facilitate zero-shot and few-shot learning techniques, which allows the model to adapt to new tasks or recognize unseen classes with limited training data. This effectively addresses the shortage of labeled datasets. GPT-3 generates high-quality pseudo-labels from minimal examples, significantly reducing manual annotation workload. While Brown et al. broadly demonstrate GPT-3's few-shot capabilities, we specifically apply GPT-3.5 to automate labeling of financial document paragraphs, directly addressing the labor-intensive labeling bottleneck in financial NLP research. Despite the benefits, we acknowledge inherent challenges like computational intensity and potential inaccuracies when handling specialized financial terminology.

Commercial NLP tools, including those built around the U.S. Securities and Exchange Commission's (SEC) EDGAR database, have become essential resources for financial analysis. These commercial platforms typically offer advanced text analysis capabilities such as semantic search, document classification, and thematic analytics. However, restrictions such as licensing costs and limited transparency significantly restrict accessibility and adaptability. Due to this, we aim to create an approach that is an open-source, transparent, and scalable NLP framework which would be accessible to a wider audience.

Methodology

We created and utilized a modularized NLP pipeline designed for parsing and analyzing SEC Form 10-K filings. Data acquisition involves fetching documents directly from the SEC EDGAR Database using a Python script (notebooks/download_10k.py). Preprocessing (preprocess.py) systematically removes non-informational boilerplate, segmenting filings into analyzable paragraphs.

AI-assisted labeling employs OpenAI's GPT-3.5 model (notebooks/ai_label.py) to automatically categorize paragraphs into classes such as financial highlights, risks, product descriptions, artificial intelligence mentions, and miscellaneous content. This streamlines the annotation process by ensuring scalability, consistency, and efficiency.

For classification, we employ logistic regression models which estimate class probabilities using a sigmoid function. This is coupled with TF-IDF vectorization, including both unigrams and bigrams (notebooks/train_logistics.py). TF-IDF features are mapped linearly to binary or multiclass labels. Additionally, we also implement a tuned BERT model for its bidirectional context capabilities and attention mechanism in hopes of higher accuracy.

Batch inference (notebooks/batch_predict.py) systematically classifies paragraphs through logistic regression and BERT models, with results structured into JSON format for comprehensive evaluation. Logistic-only predictions are generated separately (notebooks/predict_logistics.py) and directly compared against available ground truth for

detailed performance assessment. Evaluation employs precision, recall, and F1 metrics via paragraph-level alignment between predicted and true labels (notebooks/calc_scores.py).

The pipeline includes interactive spot-checking (notebooks/predict.py), enabling real-time classification and iterative model refinement. The extraction targets various financial information, such as financial statements, risks, product announcements, and AI initiatives, alongside variations including payments, assets, parties, contracts, and liabilities.

This methodology delivers a comprehensive NLP framework addressing previous limitations by providing an open, transparent, and reproducible process for reliably extracting critical information from SEC Form 10-K filings.

Implementation

We created and maintained a corpus of Form 10-K filings from the SEC EDGAR system, starting the popular American technology companies. A custom Python script (download_filings.py) was created to automate the downloading, and stored in a data directory. After downloading, each 10-K file was converted into paragraphs through the Beautiful Soup parsing library in order to strip away HTML tags, scripts, and boilerplates. This created around 25,000 paragraphs. Each paragraph is treated as an independent unit for classification, as they tend to concentrate on one particular theme. A page would be too broad, and a sentence too niche. All paragraphs were saved in a .json format.

Creating training labels has main parts. First, we used human annotation, with each team member annotating three companies. The second is using OpenAI's API to annotate each paragraph, both according to the five key target categories (financial highlights, risks, products, AI mentions, other). A script (notebook/ai_label.py) was written in order to collect each paragraph and prompt GPT-3.5 to label. The prompt is as follows:

> *Classify each of the following paragraphs using these labels:*
>
> *A financial highlight should be labeled 1. A financial highlight is often a statistic about how the company is doing. Common examples are revenue, EBITDA, year over year, etc. Keep it to these types of stats that a casual investor can*
> *view and quickly see how the company is doing.*
>
> *Risks should be labeled 2. Risks are potential dangers that will affect the company's businesses. Do not label it a risk unless it explicitly talks about a risk that will impact the company's reputation or revenue or impact.*
>
> *Products should be labeled 3. Products are simply what the company has been working on in the past year. These paragraphs should be specifically about certain products, ie. Google has youtube and Meta has instagram. Do not label as a product if unclear.*

AI should be labeled 4. Make sure it is talking about AI, or an LLM, or something of the sort. do not just quote intelligence and label that as AI.

Everything else should be labeled as 0, as that is not relevant to our user to read. Most paragraphs should be 0 unless they clearly fit one of the other categories.

Return your answer as a list of numbers separated by commas, in order. Example: 0, 1, 2, 0, 3, 0

Each JSON file produced by the script was reviewed by a human to reduce the risk of error. While GPT-3.5 provided an extremely fast process, there were times where it misclassified edge cases of ambiguous paragraphs. However, it was much more efficient to identify mislabels then to label the paragraphs manually.

Given the labeled paragraphs, we were able to transform each text into a vector using Term Frequency-Inverse Document Frequency (TF-IDF) representation. Each paragraph was encoded with weighted unigram features. Bigrams were deliberately excluded—constructing bigram features would have squared vocabulary size, and the size of the training corpus would have resulted in poor IDF estimates.

By capturing major financial and technological keywords, we can illustrate the advantages of TF-IDF: domain-specific keywords such as “EBITDA” or product names weighed much heavier than other words. Each paragraph was lowerased and tokenized, but we decided against lemmatization as the goal was to retain the information from those paragraphs as a whole.

First, for each paragraph d, we compute the term frequency:

$$TF(t,d) = (\text{count of term } t \text{ in } d)/(\text{sum of count of all vocabulary in } d)$$

Next, we compute the document frequencies and inverse document frequency over all N paragraphs:

$$DF(t) = |\{d: t \in d\}|$$
$$IDF(t) = \log(N/DF(t))$$

Finally, we can calculate the TF-IDF(t,d):

$$TF(t,d) * (IDF(t))$$

We chose a logistic regression model for paragraph classification due to its efficiency and baseline for other NLP tasks. By heavily tuning hyperparameters to maintain focus on methodology, the classifier used L2 regularization with C=1.0 strength. Training on thirty thousand examples with TF-IDF features gave the model weight vectors for each category to determine a classification decision.

Along with a TF-IDF + logistic regression classifier model, we implemented a transformer-based classifier using BERT to compare a deeper semantic analysis of the text. It was tuned on the same corpus, but ultimately used just as a gauge on whether a modern NLP model would outperform our simpler logistic regression model.

A quick analysis of our first few runs revealed the models' confusion in two distinct areas. First, the paragraphs had fewer unigrams to go off of, nothing more than section headers and the most popular financial highlights. Second, semantically rich paragraphs or numerical patterns were being glossed over. We saw an opportunity to improve our F-1 score if we targeted these two areas.

For each target category, we created a list of high-weight terms that we felt the model should bias more towards. Financial highlights now looked for more than just "EBITDA", "Revenue," and "net income," but also "earnings per share," "gross margin," and more. Risks included words not just words such as "uncertain" and "liability", but also modal verbs such as "may" and "could." Each paragraph was then scored again by the fraction of tokens in each matching list.

We also developed a small set of regex patterns to devise:

1. Monetary amounts: \$\s*\d[\d,]*
2. Percentages: \d+(\.\d+)?
3. Multiword phrases such as machine learning, artificial intelligence, etc.
4. Product terms
    a. Ex. for Google: Gemini AI, Google Drive, Google Cloud
5. Semantics
    a. Ex. "We launched": r"\b(?:in\s+\w+\s+)?we(?:\s+\w+){0,3}?\s+launched"

New features added to the TF-IDF model yielded gains in all metrics, and proved that a human touch can be added in all areas of the process.

Experiments and Results

To evaluate the algorithm, a training corpus of 18 Form 10-Ks was originally compiled, with each document containing 500-2500 paragraphs. These 18 annual reports filed with the SEC were sourced from the following companies: Airbnb, Apple, Coinbase, Datadog, Microsoft, Netflix, Nvidia, Palantir, Palo Alto Networks, Qualcomm, Salesforce, Tesla, BlackRock, Akamai Technologies, Allstate, and Axon. This data was taken from the 2024 reports from each company; this was to ensure the model was trained on the most recent information. There was a concern that older reports would have less information on new technology such as AI and would therefore not be adequate for training the model. Of the 18 Form 10-Ks used in the training corpus, 10 had labels initially generated by OpenAI's ChatGPT and subsequently reviewed by

humans, while the remaining eight, Apple, Microsoft, Nvidia, Tesla, BlackRock, Akamai Technologies, Allstate, and Axon were labeled entirely by humans.

The development set consisted of eight Form 10-Ks sourced from the following companies: Alphabet, Uber Technologies, Adobe, Amazon, Meta, AT&T, Bank of America, and Advanced Micro Devices. These were again taken from the companies' 2024 filings. Of these eight documents, four had answer keys initially generated by OpenAI's ChatGPT and subsequently reviewed by humans, while the other four, Amazon, Meta, Advanced Micro Devices, and Adobe, were labeled entirely by humans. These answer keys were used to compare against the labels produced by our algorithm for each respective document.

For the evaluation metrics, we calculated a weighted F1 score for each document in the development and test sets to assess our system's performance. The F1 score for each label would be calculated, and then the weighted F1 score would be averaged. We decided to use a weighted F1 score because it incorporates precision, recall, and accuracy for each label in a document while also accounting for the disparity between labels through support. This is important because there was a disproportionate amount of paragraphs that would be labeled as 0 (irrelevant for a casual investor), 1 (financial highlight), or 2 (risks) compared to 3 (products) or 4 (AI-related). Also, to ensure that the performance of our algorithm did not vary significantly between the AI-labeled data and human-labeled data, we ran experiments that evaluated the performance on only human-labeled training data and compared those results to the performance when using the full dataset.

We first evaluated our algorithm with the full training dataset and the development set. Using Advanced Micro Devices' Form 10-K, whose answer key was entirely labeled by humans, we compared the algorithm's predicted labels against the manually assigned labels. The document contained 919 paragraphs, of which our algorithm labeled 557 as tag 0, 125 as tag 1, 158 as tag 2, 64 as tag 3, and 15 as tag 4. The F1 score for each individual tag was 71.06%, 48.86%, 69.49%, 46.15%, and 36.00%, respectively. This evaluation yielded a weighted average F1 score of 65.46% prior to further model development. We then ran the algorithm on Alphabet's Form 10-K, whose answer key was originally labeled by AI and reviewed by humans. This document contained 1,075 paragraphs, with the algorithm assigning 666 as tag 0, 175 as tag 1, 179 as tag 2, 36 as tag 3, and 19 as tag 4. In this case, comparing the algorithm's labels to the AI-assisted answer key produced F1 scores of 72.66%, 50.00%, 77.40%, 56.45%, and 66.67% for each tag, respectively, which was used to calculate a 69.11% weighted average F1 score prior to development. This test was then repeated on the other three forms with completely human-labeled answer keys: Meta, Amazon, and Adobe. Tests on these documents received weighted average F1 scores of 62.95%, 61.04%, and 68.93%, respectively. The test was then performed on the remaining three forms with AI-assisted answer keys: Uber Technologies, AT&T, and Bank of America. Our algorithm performed similarly when compared against the AI-assisted answer keys, producing scores of 66.71%, 59.49%, and 62.55%, respectively. The same test was repeated three times for each document and always produced the same score, indicating that the results are reproducible.

After development, which involved fine-tuning our labeling system by including regular expressions that associated certain keywords with corresponding labels, we ran tests on the full training dataset and development set again. This time, for Advanced Micro Devices, our algorithm labeled 468 paragraphs as 0, 149 as 1, 198 as 2, 77 as 3, and 27 as 4. For each label, our algorithm received an F1 score of 87.64%, 82.89%, 87.32%, 78.91%, and 74.28%, respectively. Therefore, the system achieved a weighted average F1 score of 85.69%. This trend

of an improved F1 score was also reflected when we ran tests on other documents in our development set. For the Alphabet 10-K form, our algorithm now labeled 593 paragraphs as 0, 193 as 1, 204 as 2, 53 as 3, and 32 as 4. For each label, our algorithm received an F1 score of 83.47%, 78.42%, 86.93%, 74.58%, and 77.24%, respectively, which resulted in a weighted average F1 score of 82.68%. For the rest of our development set, Uber Technologies, AT&T, Bank of America, Meta, Amazon, and Adobe, our model achieved similar results, with weighted average F1 scores of 80.02%, 79.31%, 86.29%, 84.72%, and 86.83%, respectively. The same test was repeated three times for each document and always produced the same score, indicating that the results are reproducible.

Lastly, we evaluated our model's performance on the test set. Our test set consisted of four Form 10-Ks that we had entirely human-labeled keys to reference our model's performance against. The following forms were used for our test set: Qualcomm's 2001 Form 10-K, DoorDash, Accenture, and Autodesk. To ensure the validity of our results, we first conducted a test in which we only used human-labeled data. So, for this test, our training data consisted of Apple, Microsoft, Nvidia, Tesla, BlackRock, Akamai Technologies, Allstate, and Axon only. We trained our model using just this data and then predicted labels for DoorDash. Out of the 1321 paragraphs, our model predicted 643 as 0, 217 as 1, 430 as 2, 28 as 3, and 3 as 4. We compared our labels to the human-labeled DoorDash and received the following F1 scores for each label: 77.60%, 80.01%, 81.27%, 77.00%, and 59.97%, respectively. This resulted in a weighted average F1 score of 79.17%. The test on DoorDash was redone; however, the model was now trained with all the training data from before. In this test, our model labeled 667 paragraphs as 0, 215 as 1, 418 as 2, 18 as 3, and 3 as 4. The F1 score for each label was 79.04%, 81.08%, 83.27%, 79.84%, and 59.97%, respectively, which resulted in a weighted average F1 score of 80.68%. Since the majority of our test set were 2024 Form 10-Ks, to test if our model worked with older Form 10-Ks, we tested it on Qualcomm's 2001 Form 10-K. When our model was trained with only human-labeled data, it achieved a weighted average F1 score of 77.37% against our human-labeled Qualcomm 2001 form. When trained on the full dataset, it achieved a score of 79.94%. A similar test was then done on the two remaining forms in the test set. When only trained with human labels, Accenture received a weighted average F1 score of 82.21%. Then when the full dataset was used, it achieved a score of 83.60%. Similarly, when Autodesk was tested using only the human data, it achieved a score of 84.33%. Then, it received a score of 86.15% with the full dataset. When we tested with the development set added to the training corpus, for a total of 26 forms, the highest weighted average F1 score was recorded on Accenture at 87.52%. The same test was repeated three times for each document and always produced the same score, indicating that the results are reproducible.

Discussion

Throughout this research paper, we have proposed and created an algorithm that could be used to make extracting useful information from Form 10-K documents easier. Our results demonstrate the effectiveness of this approach. A related study by Cho et al. tackles a similar problem to the one presented in this research paper. Cho and his colleagues implemented a program that extracted labeled entities, such as company names and addresses, from financial documents. They incorporated the LayoutXLM model to accomplish this task. Their findings report F1 scores of 88.74% (Cho et al., 2023). While their objectives and methodology differ from ours, this indicates that our experiment's results were competitive within the field. This is evidenced by our system being able to get F1 scores of 79% - 85% when experiments on our test set were

performed using only our original training corpus, which excluded the development set. Furthermore, our system effectively reduced the volume of content that would need to be reviewed by a casual investor. For instance, in DoorDash's 2024 Form 10-K, which contains 1321 paragraphs, our algorithm filtered out 643 paragraphs (48%) as potentially unnecessary for a casual investor. This result illustrates the model's capacity to enhance document readability and focus. Therefore, it could be concluded that our research is significant to the field and that our system accomplished our goal with relative success compared to other systems in the field.

While our training set and development set answer keys include labels from ChatGPT, our results show that this did not result in a significant discrepancy in our results. For example, when we ran our system on DoorDash's 2024 Form 10-K and only used human-labeled forms to train our data, DoorDash achieved similar results to when we performed the same test but with the AI-assisted labels in the training set as well. Despite our system performing differently when the AI-assisted labeled forms were added to the training set, this can be attributed to the model being trained with more data. Also, when we compared our program's performance to human-labeled answer keys and then AI-labeled answer keys during development, there was little variation in F1 score. However, we must conduct more experiments to verify that the AI labels did not impact our data.

Limitations in our research arise in our dataset. It would be advantageous to expand our training corpus to improve the accuracy of our model, as evidenced by an increase in F1 score when the development set was added to the training corpus. Also, if adequate time, circumstances, and resources are provided, OpenAI's ChatGPT should be completely removed from our labeling system to ensure the accuracy of the training data and labels that we compare our output to. However, due to time constraints, we needed to incorporate ChatGPT to streamline the process. Furthermore, our research would benefit from conducting tests on a large and diverse test set. In our evaluation, we only incorporated one Form 10-K that was not from 2024. The F1 score that the test on the Qualcomm 2001 form was similar to our other tests, so it indicated that our model was still effective on older forms. However, if our tests are reproduced, we must incorporate older Form 10-Ks in our training corpus and more in our test set for a more adequate sample. We should also diversify the companies with more representation in fields like materials and industrials.

Conclusion

This research introduces an integrated NLP framework for extracting and classifying key financial insights from SEC Form 10-K filings, combining traditional methods like TF-IDF with advanced Transformer models such as BERT. The system demonstrated relative accuracy to other works, usually achieving F1 scores between 79% and 85%, which indicates that the system is able to accurately highlight key information for a casual investor. Labeling paragraphs in financial documents using a BERT model is significant to the field of NLP because it demonstrates the practical application of deep contextual language models to complex text. Financial documents pose unique challenges due to their length, formal tone, and specialized vocabulary, making them valuable examples for advancing NLP techniques. This work showcases how BERT can be fine-tuned to perform classification tasks in real-world applications.

While the results are promising, there are clear opportunities for future experiments. Future experiments should add more labels for finer classification. If possible, the labels should be used to create a comprehensive summary, further condensing the necessary information. We

call on future researchers to build on this work, refine the methodology with advanced models, and explore new avenues for advancing financial document analysis. Together, we can push the boundaries of making critical financial data more accessible.

Contributions Breakdown

The group initially chose to explore a research project focused on tokenizing sections of legal documents, with the aim of highlighting key portions and presenting them in a comprehensible manner. After deliberation within the group, Christopher Li proposed that the project be built around tokenizing Form 10-K financial documents from companies, forms in which publicly traded companies extensively disclose their endeavors for a given year. The group then collectively discussed how this could be accomplished. We first thought to label each sentence of a Form 10-K with a tag; however, we ultimately decided to label each paragraph of a Form 10-K with the ones mentioned in the implementation and results. In the meantime, Sam Murshed, the group's primary researcher, explored past works and suggested an implementation of the BERT model since it was used in similar works.

Christopher and Howard Appel, the primary programmers, successfully programmed a baseline algorithm, with the assistance of Sam Murshed and Jaylon McDuffie. The entire group contributed to searching for potential documents to include in a training corpus and use for testing. The program would label documents with prompts to OpenAI's ChatGPT; these labeled documents would then be equally divided amongst and manually checked by the group to see if we agreed with the labels.

The partly AI and partly human-labeled documents would serve as part of the training corpus and answer keys to assess the performance of the BERT model implementation on the development and test sets. To ensure accuracy, each group member manually labeled four Form 10-Ks; these 16 entirely human-labeled documents would be divided amongst the training, development, and test sets. Howard, the primary performance analyst, reran all the tests originally performed by Christopher, who conducted initial evaluations to ensure the model was functioning correctly. Howard worked with the development set and reported the results of the baseline program to the rest of the group. Howard then worked closely with the rest of the group to improve the system's results on the development set.

All of the group members contributed to presenting the research in New York University's Spring 2025 Natural Language Processing course; Jaylon introduced the topic and provided context, Sam discussed the previous works that inspired this topic, Christopher discussed the methodology, and Howard presented the results and evaluation metrics. All of the group members also contributed to writing this paper; Jaylon composed the abstract/introduction, Sam presented his research in the literature review and how that research influenced the methodology, Christopher articulated the implementation of the system, and Howard formulated the contributions and results/discussion sections.

References

Cho, S., Moon, J., Bae, J., Kim, J., & Lee, S. (2023). A framework for understanding unstructured financial documents using RPA and multimodal approach. *Electronics*, *12*(4), 939. https://doi.org/10.3390/electronics12040939

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

Brown, T., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513-523.

Shannon, C. E. (1951). Prediction and Entropy of Printed English. Bell System Technical Journal, 30(1), 50-64.