IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 January 2024, accepted 22 February 2024, date of publication 26 February 2024, date of current version 4 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3370444

APPLIED RESEARCH

Language Models Fine-Tuning for Automatic Format Reconstruction of SEC Financial Filings

GIANFRANCO LOMBARDO, GIUSEPPE TRIMIGNO, MATTIA PELLEGRINO, AND STEFANO CAGNONI, (Senior Member, IEEE)
Department of Engineering and Architecture, University of Parma, 43125 Parma, Italy
Corresponding author: Gianfranco Lombardo (gianfranco.lombardo@unipr.it)

ABSTRACT The analysis of financial reports is a crucial task for investors and regulators, especially the mandatory annual reports (10-K) required by the SEC (Securities and Exchange Commission) that provide crucial information about a public company in the American stock market. Although SEC suggests a specific document format to standardize and simplify the analysis, in recent years, several companies have introduced their own format and organization of the contents, making human-based and automatic knowledge extraction inherently more difficult. In this research work, we investigate different Neural language models based on Transformer networks (Bidirectional recurrence-based, Autoregressive-based, and Autoencoders-based approaches) to automatically reconstruct an SEC-like format of the documents as a multi-class classification task with 18 classes at the sentence level. In particular, we propose a Bidirectional fine-tuning procedure to specialize pre-trained language models on this task. We propose and make the resulting novel transformer model, named SEC-former, publicly available to deal with this task. We evaluate SEC-former in three different scenarios: 1) in terms of topic detection performances; 2) in terms of document similarity (TF-IDF Bag-of-words and Doc2Vec) achieved with respect to original and trustable financial reports since this operation is leveraged for portfolio optimization tasks; and 3) testing the model in a real use-case scenario related to a public company that does not respect the SEC format but provides a human-supervised reference to reconstruct it.

INDEX TERMS SEC filings, deep learning, transformer, document reconstruction, topic detection, stock market.

I. INTRODUCTION

Reviewing SEC (Securities and Exchange Commission) filings are important for investors as they provide crucial information about a public company, such as its financial performance, business operations, and risk factors. In this way, investors can make informed decisions about whether to buy, hold, or sell a company's securities and obtain important disclosures about a company's financial condition and business operations. One of the most important documents is the annual report (10-K filing), which provides a better understanding of a company's financial strength, competitive position, and growth prospects ([1]). In light of this, several research works have demonstrated the importance of automatically analyzing 10-Ks to perform knowledge extraction, measuring the year-over-year changes between two consecutive annual filings, or detecting the sentiment reported in the most important sections. Published in 1998 by the Securities and Exchange Commission, the Plain English Handbook was the first publication providing guidelines to help public companies create clear SEC disclosure documents. This publication and the Sarbanes–Onley Act of 2002, which was constructed to supervise financial reporting, have made corporate filings an increasingly reliable source of information.

SEC suggests a document format with a specific division of topics into 20 different items of the document to improve readability and analysis. Nevertheless, several companies in

The associate editor coordinating the review of this manuscript and approving it for publication was Agostino Forestiero.

the last years have introduced their own format and organization of the contents. This condition makes human-based and automatic analysis inherently more difficult because the documents become longer and more complex every year. Moreover, large-scale content retrieval from SEC filings can be tough since reports are published in different formats, which are often unstructured or semi-semistructured. The main contributions of this research work are the following:

- We investigate different language models based on transformers neural networks to automatically analyze each paragraph in an SEC annual report and reconstruct an SEC-like format of the document as a multi-class classification task with 18 classes.
- We propose a bidirectional fine-tuning procedure of the pre-trained transformer models for this specific task based on different combinations of Bi-Directional Long-Short Term Memory Networks (Bi-LSTMs)
- We made publicly available the best model achieved, named SEC-former.
- We also evaluate the results of SEC-former in two different use cases: a) Document similarity detection for portfolio optimization and b) A qualitative reconstruction of a real document that is not aligned with the SEC standard format.

The paper is structured as follows: Section II presents a literature review of Natural Language Processing techniques that exploit information from SEC filings for different financial applications; Section III describes the Data Collection process and the composition of the dataset; Section IV introduces the operations performed on data collected from SEC's EDGAR platform to build different datasets; Section V presents the methodology adopted for the Document Format Reconstruction task; Section VI presents three different kinds of results for the Document Format Reconstruction task; the paper ends with a concluding evaluation of the research work.

II. RELATED WORKS

Natural Language Processing (NLP) techniques have been widely applied in the financial domain in the last four decades. Financial news, in particular, has been extensively exploited to make market predictions. Social media and corporate disclosures are also being increasingly utilized in various applications. Reference [2], for example, produced a multi-layer algorithm composed of different machine learning models that exploit the semantics and sentiment of news headlines for a FOREX market prediction task. Reference [3] proposed a novel fine-grained approach that captures the explicit and implicit topic-dependent sentiment in company-specific news text. Market volatility related to financial news has also been studied extensively by [4] with neural networks (NNs).

More recently, annual SEC reports have attracted more interest from both investors and researchers, especially thanks to the research work "Lazy Prices" by [5], which argued that a simple comparison of consecutive 10-Ks hides a lot of valuable information. Indeed, L. Cohen et al. showed that the firms' management is often "lazy" and uses the last year's filings verbatim in constructing the current year's 10-K while making only the necessary changes to be within the boundaries of fiduciary responsibility. Observing these changes yields an important and robust indication for future firm performance. They demonstrate this assumption by computing quintiles from the distribution of the similarity scores from all companies' filings, based on which they construct equally weighted and capitalization-weighted portfolios to prove that this phenomenon affects the entire Stock market. Specifically, they show that buying stocks of the "non-changers" and short-selling the "changers" yields statistically significant abnormal returns. In light of this, breaks from previous standardized reporting can significantly affect firms' future stock returns. However, the Lazy prices' methodology is based on Bag-Of-Words and TF-IDF encoding of the documents, which ignore semantically similar words and suffer from sparse-encoding representations of the documents. According to [6], it is normal for managers to be incentivized to minimize (maximize) the effects on their companies' stock prices deriving from negative (positive) news about their firms. Reference [7] showed that the managers provide boilerplate information and avoid giving accurate hints about the company's status by extending the document length. However, the SEC prohibits any misleading statement or omission under Rule 10b-5 and requires that a company's CEO and CFO certify the accuracy of the 10-K. This means that, even though valuable information about the company and the industry does exist in the 10-Ks, the management has incentives to hide it.

To deal with these issues, [8] proposes the analysis of semantic similarity changes among consecutive annual SEC reports by exploiting neural networks' dense representations; in particular, Doc2Vec ([9]), for decision-making in portfolio optimization. A similar method has also been applied by [10] to detect bank distress by mining financial news. As well, extracting information from annual reports has recently gained more and more attention in the credit risk sector since valuable information to estimate the probability of a firm's default can be extracted from 10-Ks. Reference [11] proposed a deep learning model that combines document embedding and Convolutional Neural Networks to predict the probability of bankruptcy by merging accounting variables and text extracted from Item 7 (Management Discussion) of 10-k annual reports. More recently, state-of-the-art Transformer networks have also been used to extract information from SEC filings. Reference [12] leverage pre-trained transformers to extract sentiment and topic from 10-Ks to predict companies' performance over the next year, also providing explainable results for the downstream performance prediction task. Despite the increasing interest of the AI and financial research communities, all the methods previously presented have a common limitation in real-case scenarios related to data quality. Indeed, in recent years, it has become particularly difficult to analyze SEC filings automatically because of the heterogeneous structure of the

reports adopted by several companies. In many cases, they differ only slightly from the regular SEC filings but can also be definitely different. Moreover, although SEC filings are publicly available from the SEC EDGAR platform, they are often released with different formats that complicate the extraction of the text content and the correct segmentation of the document into the appropriate thematic items.

III. NEURAL LANGUAGE MODELS

In this section, we revise the main features of the State-of-the-Art Neural Language Models we exploited for our analysis and to design the proposed model for the Document Format Reconstruction task. All the models are pre-trained Transformers networks [13] that rely on different self-attention mechanisms to generate a contextualized embedding of the input text. Transformer-based models usually exploit an encoder component that learns a representation of words along with some special tokens and a subsequent neural network that is specialized on a specific NLP task or a decoder in the case of Text Generation tasks.

A. BERT

BERT (Bidirectional Encoder Representations from Transformers, [14]) is a pre-trained transformer-based model that obtained state-of-the-art results in a wide variety of NLP tasks, such as question answering (SQuAD dataset, [15]) and natural language understanding (GLUE benchmark, [16]). While Recurrent Neural Networks process texts as a sequence of single words by exploiting a recurrent mechanism, BERT exploits a self-attention mechanism based on several attention-heads, which process the entire text independently with a Query-Value system that identifies the relationships among words in the text. In light of this, each word's context is totally observable for the model that, consequently, can generate contextual embeddings. Moreover, BERT introduces special tokens to characterize the structure of each sentence, such as `[CLS]` **x** `[SEP]`. A BERT model is characterized by: a) The number of Transformer blocks as L, every block is a module including an encoder network with a self-attention mechanism; b) the hidden size, denoted as H, representing the embedding dimension of each sentence; c) The number of self-attention heads A; d) The model size, expressed as the total number P of parameters/weights.

BERT is pre-trained on the Wikipedia and BookCorpus datasets ([17]) for two unsupervised tasks:

- **Masked LM** In this task, 15% of all input tokens are selected at random for replacement with a special `[MASK]` token. 80% are actually replaced, 10% are left unchanged, and a randomly selected token replaces the remaining 10%. The model has to predict the original value of the masked words.
- **Next Sentence Prediction (NSP)** Many downstream tasks, such as Question Answering, are based on understanding the relationship between two sentences. In this task, the model receives pairs of sentences as input (with a `[SEP]` token after the first one and a `[CLS]` token after the second one), and it has to predict whether the second sentence is the subsequent sentence in the original document.

In the section describing the experimental results, we report results obtained by two models: **BERT$_{\mathbf{BASE}}$** (L=12, H=768, A=12, P=110M) and **BERT$_{\mathbf{LARGE}}$** (L=24, H=1024, A=16, P=340M), both *uncased* version.

B. ROBERTA

RoBERTa (Robustly Optimized BERT Approach) is a modified version of BERT proposed by [18], which includes (1) longer training, with more data and larger batches; (2) removing the NSP pre-training task; (3) The ability to process longer sequences; (4) dynamic masking over training data.

In addition to the BookCorpus and Wikipedia datasets, three other corpora are used to pre-train the model: CC-News [19], OpenWebText [20], and Stories.

The most important change is dynamic instead of static masking. Dynamic masking generates a new masking pattern every time a sequence is fed into the model, while BERT performs masking only once during data pre-processing (static masking).

Moreover, the training data were duplicated 10 times so that each sequence was masked in 10 different ways over the training phase.

In our experimental part, we report results on this model having the following configuration: **RoBERTa$_{\mathbf{BASE}}$** (L=12, H=768, A=12, P=110M), which is *cased*.

C. XLNET

XLNet is a generalized auto-regressive pre-training method proposed by ([21]) and based on the Transformer-XL ([22]). It is designed to overcome the limitations of BERT by using a permutation-based training method, which allows it to better capture the dependencies between all words in a sentence, as opposed to the masked language modeling objective used in BERT, which only considers dependencies within a local context. This is achieved by training the model to predict a randomly permuted version of the input sequence rather than just predicting missing tokens in the original sequence.

Another key difference between XLNet and BERT is that XLNet uses a segment-level recurrent mechanism, which allows it to better model the dependencies between different segments in the input sequence. This is especially useful in tasks such as text classification, where it is important to understand the relationships between different sentences or paragraphs.

In terms of architecture, XLNet is similar to the Transformer architecture used in BERT, with a few modifications. It uses self-attention mechanisms to calculate the representations of each word in the input sequence and feed-forward networks to refine these representations. Overall, XLNet has achieved state-of-the-art results on a number of NLP

tasks, including text classification, sentiment analysis, and question answering, due to its improved ability to capture dependencies between all tokens in a sentence.

The idea behind XLNet is to leverage the best of both auto-regressive (AR) language modeling and auto-encoding (AE) while avoiding their limitations. It achieves bidirectional contextual embedding of the input sequence by maximizing the expected likelihood over all input sequence factorization order permutations, as explained in the following.

AR language modeling seeks to estimate the probability distribution of a text corpus with an auto-regressive mechanism that, given an input sequence $x = [x_1, \ldots, x_N]$, maximizes the likelihood of a forward product (Eq. 1) or a backward product (Eq. 2), where p_θ is achieved by training a parametric model (e.g., a neural network).

$$p_\theta(x) = \prod_{t=1}^{T} p_\theta(x_t|x_{<t}) \tag{1}$$

$$p_\theta(x) = \prod_{t=T}^{1} p_\theta(x_t|x_{>t}) \tag{2}$$

Considering the forward product, to optimize the log probability of the text corpus distribution, the AR language modeling optimization task can be described as in Eq. 3:

$$\begin{aligned}\max_\theta \log p_\theta(x) &= \sum_{t=1}^{T} \log p_\theta(x_t|x_{<t}) \\ &= \sum_{t=1}^{T} \log \frac{\exp(h_\theta(x_{1:t-1}))^\intercal e(x_t))}{\sum_{x'} \exp(h_\theta(x_{1:t-1}))^\intercal e(x'))}\end{aligned} \tag{3}$$

where $h_\theta(x_{1:t-1})$ is the context representation produced by the neural models (considering only tokens to the left), and $e(x_t)$ is the embedding of each word x_t.

On the other hand, BERT-like models are based on a denoising auto-encoder architecture that constructs a corrupted version $\hat{x}$ by masking words with the `[MASK]` token. The AE language modeling training objective is to reconstruct masked tokens $\bar{x}$ from $\hat{x}$, so:

$$\begin{aligned}\max_\theta \log p_\theta(\bar{x}|\hat{x}) &\approx \sum_{t=1}^{T} m_t \log p_\theta(x_t|\hat{x}) \\ &= \sum_{t=1}^{T} \log \frac{\exp(H_\theta(\hat{x}))_t^\intercal e(x_t))}{\sum_{x'} \exp(H_\theta(\hat{x}))_t^\intercal e(x'))}\end{aligned} \tag{4}$$

where m_t is 1 if x_t is masked, and $H_\theta(\hat{x})$ is a Transformer network that maps a fixed length-T input sequence $\hat{x}$ into a sequence of T hidden vectors $[H_\theta(\hat{x})_1, \ldots, H_\theta(\hat{x})_T]$.

AR language modeling is limited to uni-directional contexts (forward o backward), while AE language modeling shows two limitations: (1) it factorizes the joint conditional probability $p_\theta(\bar{x}|\hat{x})$ based on an independence assumption according to which all masked tokens are separately reconstructed (which justifies the $\approx$ sign in Eq. 4); (2) special tokens, as `[MASK]`, never occur in downstream tasks, creating a pre-train fine-tune discrepancy.

To merge the two approaches and overcome the limitations, XLNet exploits the **Permutation Language Modeling** (PLM) objective ([23]) that enjoys the benefits of AR models while capturing a bidirectional context. For a sequence, **x** of length T, AR factorization can be performed in $T!$ different orders. So, if model parameters are shared across all factorization orders, the model will learn information from all positions in both directions.

Let us call $\mathcal{Z}_T$ the set of all possible permutations of length-T index sequence, z_t the t^{th} element and $z_{<t}$ the first $t-1$ elements of $z \in \mathcal{Z}_T$. The PLM objective can be expressed as:

$$\max_\theta \mathbb{E}_{z\sim\mathcal{Z}_T} \left[\sum_{t=1}^{T} \log p_\theta(x_{z_t}|x_{z_{<t}}) \right] \tag{5}$$

This objective formulation only permutes the factorization order, not the sequence order. However, this objective formulation causes slow convergence. For this reason, XLNet limits the prediction task only to the last tokens in the factorization order (partial prediction). We address the reader to [23], and [21] for more details.

As for pre-training data, XLNet uses BookCorpus and Wikipedia, along with the following datasets: Giga5, ClueWeb, and CommonCrawl.

In Section VI, we report results obtained using the following model: $XLNet_{LARGE}$ (L=24, H=1024, A=16, P=340M).

IV. DATA

This section introduces the data we collected from the EDGAR SEC platform and the operations performed to build different datasets. In particular, we collected 10-K annual reports released between 2011 and 2022 of 6k public companies traded in the American stock market (New York Stock Exchange and Nasdaq). According to SEC recommendations, a 10-K document should be arranged in four parts and 20 thematic items (see Table 1). Each item contains an arbitrary number of paragraphs along with tables

TABLE 1. Structure suggested by SEC for the annual report (10-K).

Part 1	
Item 1	Business
Item 1A	Risk Factors
Item 1B	Unresolved Staff Comments
Item 2	Properties
Item 3	Legal Proceedings
Item 4	Mine Safety Disclosures
Part 2	
Item 5	Market
Item 6	Consolidated Financial Data
Item 7	Management's Discussion and Analysis of Financial Condition and Results of Operations
Item 7A	Quantitative and Qualitative Disclosures about Market Risks
Item 8	Financial Statements
Item 9	Changes in and Disagreements With Accountants on Accounting and Financial Disclosure
Item 9A	Controls and Procedures
Item 9B	Other Information
Part 3	
Item 10	Directors, Executive Officers, and Corporate Governance
Item 11	Executive Compensation
Item 12	Security Ownership of Certain Beneficial Owners and Management and Related Stockholder Matters
Item 13	Certain Relationships and Related Transactions, and Director Independence
Item 14	Principal Accounting Fees and Services
Part 4	
Item 15	Exhibits, Financial Statement Schedules Signatures

and graphs. All the images and tables are deleted by filtering the HTML content of the document.

A. DATASET FOR 10-K FORMAT RECONSTRUCTION

We implemented the 10-K format reconstruction task as a multi-class classification task over the paragraphs of a document. The goal is to classify each paragraph as part of one of the possible 18 items (Item 1b and Item 14 have not been considered because they are often not filled in by the companies in their reports). An example of the task is presented in Figure (1). We collected all the reports from American companies published between 2011 and 2022. We built the dataset considering only those documents where it was possible to identify each thematic item separately. This identification is necessary to label each text content to the respective thematic item according to the SEC structure (Table 1). Data labeling is performed by leveraging Regex rules on specific keywords or by analyzing the Table of Contents (when available). This filtering is performed over the text content after the HTML tags have been removed. Indeed, identifying each item in a 10-K document is not trivial due to the totally unstructured format of the original documents. Three different parsers have been developed to recover the document's division into items depending on the document structure. We have exploited the following three information sources for parsing the text:

1) The presence of a table of contents with hyperlinks in the document.
2) The search of specific keywords for each item along with a table of contents available without hyperlinks.
3) Shallow identification, by only considering the presence of consecutive keywords in the document for each item and looking for the presumed subsequent item.

There is no comprehensive solution that can parse each document because of the custom document structure adopted by each company. For this reason, we considered only those documents eligible for inclusion in the dataset for which it was possible to retrieve all the items with little "noise" (surrounding unrelated information before or after each item). However, since some items are systematically missing because they were added as attachments in other reports, and considering the high imbalance of the data set (Figure (2)), we removed all paragraphs related to Item 1B and Item 14.

The final dataset is composed of 43608 documents, each corresponding to a complete 10-K with all its items. By extracting paragraphs according to the HTML structure of the document or considering groups of sentences separated by "newline" characters, we collected a highly imbalanced dataset composed of ∼9.54M paragraphs (see Figure (2) for item distribution). Each paragraph has from 168 to 223 words, with a mean of about 200 words per paragraph. Figure (3) shows the distribution of the paragraphs over the years, and it is in line with the trend analyzed by [7]: over the years, the documents' length has been increasing, with longer and more numerous paragraphs.

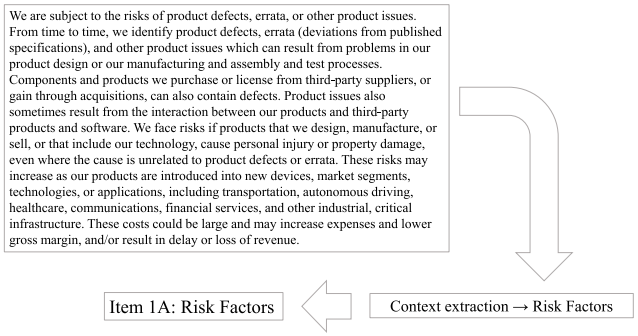


FIGURE 1. Paragraph classification based on contextual information.

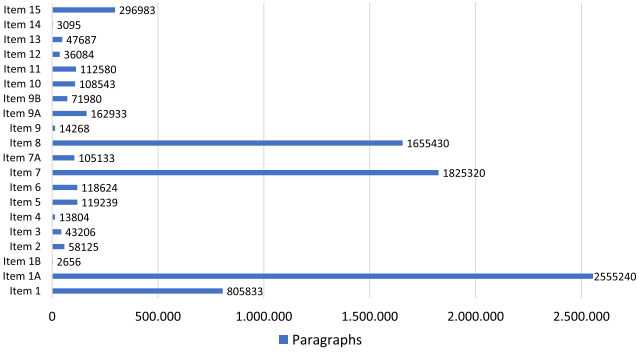


FIGURE 2. Items distribution of raw paragraphs.

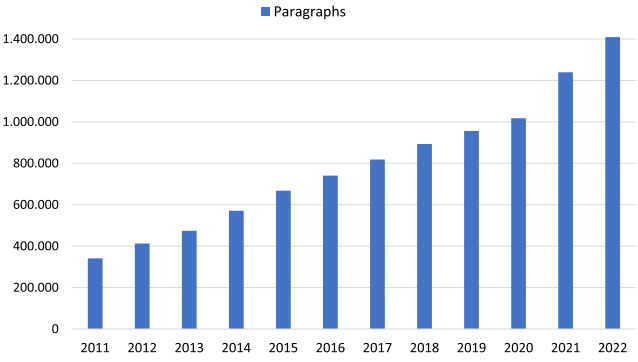


FIGURE 3. Yearly distribution of raw paragraphs.

B. DATASET COMPOSITION

In order to fine-tune, train, validate, and test the Machine Learning and Deep Learning models considered in our research, we divided the dataset according to several policies related to the following issues:

- **Mutual exclusion of companies**: Although the dataset covers a period of eleven years, the data splitting should avoid dependencies among the training, validation, and test sets caused by a strong overlapping of the companies analyzed in each year. Considering the result presented by [5] about the strong similarities between consecutive 10-Ks (copy&paste issue), we must avoid including instances (documents) of the same company from different years at the same time in the training,

validation, and test sets. To meet this requirement, data splitting is firstly done based on firms instead of time or number of paragraphs.

- **Temporal distribution of paragraphs**: According to the yearly distribution of paragraphs, it is clear that documents coming from different companies and different years have a different average length, and therefore a data splitting considering only the time variable could lead to a different distribution of the 10-Ks' items in the training, test, and validation sets.

According to the above considerations, we used paragraphs from documents between 2011 and 2021 for training, validation, and testing. Furthermore, we built a second separate test set including only paragraphs from documents published in 2022 to finally analyze a separate context able to simulate a real-case scenario for our tasks. No firm is present in more than one dataset. Since all the datasets suffer from a heavy imbalance condition among the items (samples for each predictable class), we have also performed a random under-sampling to create balanced training, validation, and test sets for further evaluation. In light of this, the final composition of the dataset is the following:

- **Training set**: 13,500 paragraphs per item (for a total of 243,000 paragraphs), related to 3,709 companies.
- **Imbalanced Validation set**: 77,043 paragraphs (related to 531 companies), distributed as in Figure (4).
- **Balanced validation set**: after under-sampling, it contains 1,234 paragraphs per item.
- **Imbalanced test set**: 151,880 paragraphs, related to 1,060 companies, whose distribution is shown in Figure (4).
- **Balanced test set**: after under-sampling, it contains 2,771 paragraphs per item.
- **Test set (2022)**: 765,315 paragraphs from 2022, distributed as in Figure (5).

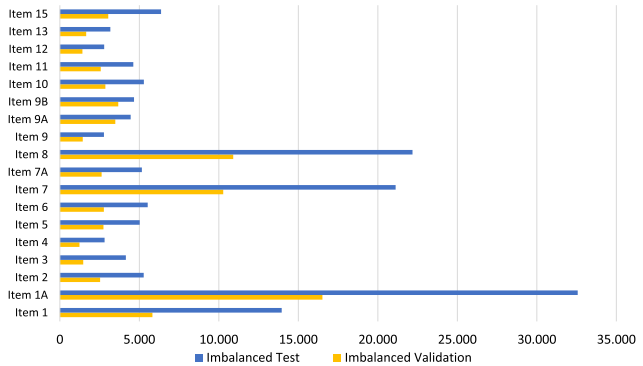


FIGURE 4. Paragraph distribution for imbalanced validation and test set.

C. DATA PRE-PROCESSING

In our experiments, we have investigated and compared several classes of models for the Document Format Reconstruction task. We investigated different transformer-based solutions by proposing different fine-tuning techniques.

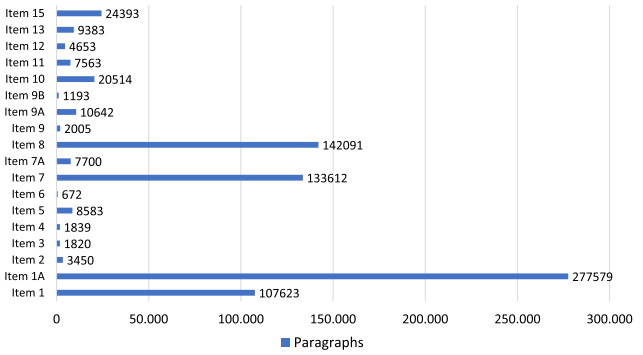


FIGURE 5. Paragraphs distribution for the year 2022.

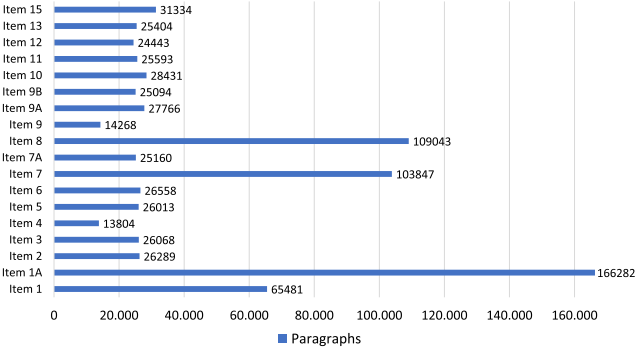


FIGURE 6. Item distribution for years 2011-2021.

We compared the results with two baseline models, XGBoost and a Bidirectional LSTM to prove the benefits introduced by our proposed solutions. In light of this, it was necessary to build different pre-processing pipelines to perform our experiments, depending on the class of models considered. Indeed, transformer-based models require few pre-processing steps (e.g., data cleaning) since the models are pre-trained over large text corpora; thus, the transformers already provide an initial word embedding for most words ([24]). On the other hand, baseline ML models like XGBoost and LSTM benefit from the typical NLP pre-processing pipelines to reduce the number of features, such as Stemming and Stopword Removal. In light of this, to enhance the repeatability of our experiments, we present the pre-processing tasks divided into two sets:

- **Transformer-oriented tasks**: Operations limited to the data tokenization tasks required by each pre-trained model. In particular, we used the *BertTokenizer*,[1] the *RobertaTokenizer*,[2] and the *XLNetTokenizer*[3] with the same sentence padding for each model, considering the maximum length equal to 200 and truncating longer paragraphs.
- **BiLSTM & XGBoost-oriented tasks**: documents are encoded according to the *TF-Idf* (Term Frequency

[1]Refer to official Huggingface's BertTokenizer documentation.
[2]Refer to official Huggingface's RobertaTokenizer documentation.
[3]Refer to official Huggingface's XLNetTokenizer documentation.

Inverse Document Frequency) statistic and using the Snowball stemmer.

1) BILSTM & XGBOOST-ORIENTED PRE-PROCESSING TASKS

We removed punctuation, stop-words, and any non-English words to let the model focus on the most relevant words. Finally, we applied the Snowball stemmer. To encode paragraphs for XGBoost, we exploited the *Tf-Idf* statistic to reflect each word's relevance, considering the paragraph to which it belongs and the entire collection of 10-Ks as shown in equations (6) and (7).

$$\text{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$
$$\text{idf}(t,D) = log\frac{f_{t,d}}{|\{d \in D : t \in d\}|} \tag{6}$$
$$\text{tf} \cdot \text{idf}(t,d,D) = \text{tf}(t,d) \cdot \text{idf}(t,D). \tag{7}$$

where $f_{t,d}$ is the number of occurrences of word t in paragraph d, $\sum_{t' \in d} f_{t',d}$ is the number of words in paragraph d and $|\{d \in D : t \in d\}|$ is the number of paragraphs which contain word d at least once. Moreover, considering the amount of memory required to manage the vocabulary accumulated through all the documents and in order to speed up the models' training, we selected an occurrence threshold equal to 7. This way, we built a vocabulary of size 11,952, i.e., each paragraph will be represented as a vector of size 11,952.

Instead, for the Bi-LSTM, we opted for **Keras tokenizer** considering a 20,000-word vocabulary, which returns an initial word embedding for each token.

V. METHODOLOGY

This section presents the methodology we followed to investigate the Document Format Reconstruction task. We describe the metrics we considered to design the models and to evaluate them on the final test sets. We describe the fine-tuning mechanisms we propose to specialize the transformer architectures for this task. Finally, we also present the two baseline models based on XGBoost ([25]) and Bidirectional LSTM (BiLSTM- [26]) to prove the benefits of our solutions.

A. EVALUATION METRICS

Since the Document Format Reconstruction task involves a multi-class classification task with an imbalanced distribution over the 18 selected classes, our overall evaluation is based on different evaluation metrics. For the balanced validation and test sets, we refer to the **Accuracy**, defined as the ratio of correctly predicted samples over the total ones.

For the imbalanced validation and test sets, we computed the **Precision**, the **Recall**, and the **F1-Score**, defined as the harmonic mean of Precision and Recall. We also considered the Area Under the Curve (**AUC**), which measures the ability of a classifier to distinguish between classes and is used as a summary of the Receiver Operating Characteristic (ROC) Curve. The ROC curve is created by plotting the true positive rate (**TPR**) against the false positive rate (**FPR**) at various threshold settings.

B. BASELINE MODELS

The first baseline model we considered is the Extreme Gradient Boosting (XGBoost) algorithm. It is an ensemble learning mechanism that leverages decision trees trained sequentially on the residuals (errors) of the previous model. This model has been selected because of the performance shown in several applications and since it is considered a valid alternative to Deep Learning solutions, especially for financial-oriented tasks (see [27], [28], [29]). For a fair comparison, we optimized the XGBoost model with a grid-search procedure leveraging the validation set. In particular, the best performance was achieved with the following setting: 500 estimators, learning-rate=$1e^{-6}$, max-depth=11, min-child-weight=5, reg-lambda=1.0, reg-alpha=$1e^{-4}$, gamma=0.1. The XGBoost architecture is shown in Figure 7.

As the second baseline model, we considered the Bidirectional Long Short-Term Memory Network (Bi-LSTM) since it can process a text input both from left to right and from right to left while optimizing the embedding of each sentence during training. The Bi-LSTM works as an encoder network to learn the context embedding of each sentence and requires an additional stack of feed-forward layers (DNN) to specialize the embedding for the classification task needed for the Document Format Reconstruction task. The final structure of this network has been defined after a grid-search optimization that involved 50 different combinations of the Bi-LSTM and the DNN layers evaluated over the validation set. In particular, we considered different design choices, such as leveraging only one Bi-LSTM or a stack of two Bi-LSTMs. The final architecture includes a) two stacked BiLSTMs with 96 units each and a dropout layer after each LSTM unit; b) a feed-forward network with two layers with 64 and 32 hidden neurons, respectively. As for the hyper-parameters, the best results have been achieved with batch size=64, learning-rate=$6e^{-3}$. The network has been trained by applying the Early-Stopping technique to avoid

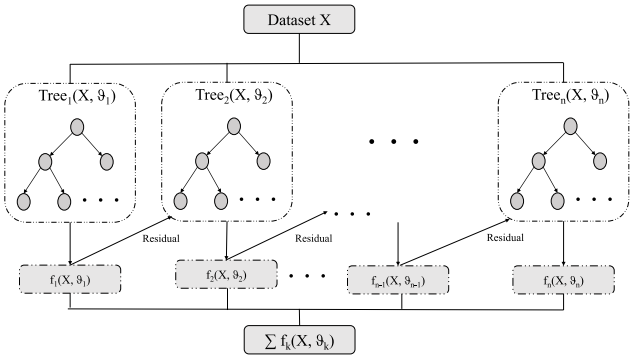


FIGURE 7. Baseline models: XGBoost architecture.

TABLE 2. Results on the balanced and imbalanced validation sets.

Model	Balanced validation set	Imbalanced validation set			
	Accuracy	Precision	Recall	F1-score	AUC
Random choice	0.056	-	-	-	-
XGBoost	0.611	0.596	0.565	0.57	0.88
BiLSTM + DNN	0.702	0.709	0.687	0.689	0.947
BERT-base + DNN	0.785	0.793	0.779	0.781	0.973
BERT-base + BiLSTM + DNN	0.784	0.791	0.777	0.778	0.972
BERT-large + DNN	0.788	0.790	0.779	0.781	0.973
BERT-large + BiLSTM + DNN	0.787	0.789	0.782	0.782	0.974
FinBERT + DNN	0.741	0.747	0.730	0.732	0.960
FinBERT + BiLSTM + DNN	0.739	0.745	0.731	0.733	0.961
RoBERTa + DNN	0.748	0.745	0.730	0.733	0.961
RoBERTa + BiLSTM + DNN	0.781	0.778	0.771	0.773	0.972
XLNet-large + DNN	0.755	0.761	0.746	0.746	0.966
XLNet-large + BiLSTM + DNN	**0.793**	**0.805**	**0.791**	**0.793**	**0.976**

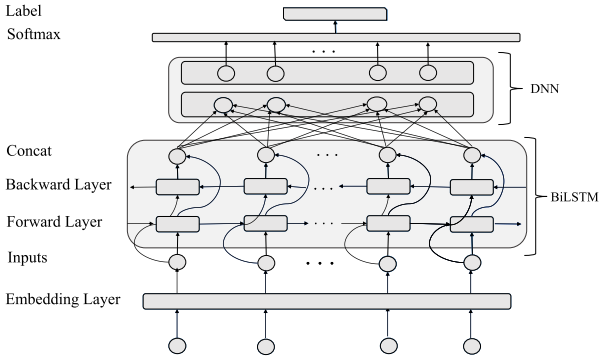


FIGURE 8. Baseline models: BiLSTM + DNN architecture.

overfitting and by leveraging the Categorical Cross-Entropy loss function. The architecture is shown in Figure 8.

Table 2 shows the results achieved with the best models over both the balanced and the imbalanced validation sets.

C. TRANSFORMERS' FINE-TUNING

For the Document Format Reconstruction task, we considered the transformer networks introduced in Section III. Since all these networks are available as models pre-trained over different tasks and data sets, we propose a fine-tuning procedure for our specific task based on our data set. According to the number of attention-heads, each model can be defined as "base" or "large". Increasing the number of attention heads usually leads to better performance, although with a higher computational cost due to a proportional growth of the weights in the network. The main benefit of relying on pre-trained models is the possibility to exploit the capability acquired by each model to represent most of the words in different domains according to a common context, therefore not limited to the typical financial jargon. Aiming to investigate this last point in particular, we considered BERT, RoBERTa, and XLNet models in both their "base" and "large" models, which are pre-trained without any specific exposure to the financial jargon. On the other hand, to evaluate the benefits of the exposure to the financial terms, we considered FinBERT ([30]), which is a BERT-base model that has been fine-tuned for sentiment analysis of financial documents on the following financial data sets:

- The **Financial PhraseBank**, sentences selected randomly from financial news found on the LexisNexis database, then annotated by 16 people with backgrounds in finance and business. The annotators were asked to assign labels according to the extent to which they thought the information in the sentence might affect the mentioned company's stock price.
- A subset of the **TRC2 Reuters news** dataset that includes only financial news.

For each pre-trained model, we removed the last task-specific layers and fine-tuned the model for the Document Format Reconstruction task by mainly evaluating two architectural strategies:

1) **DNN fine-tuning**: fine-tuning using a Deep Feed-Forward network that is directly fed with all the token embeddings (including the [CLS] token) produced by the transformer network (Figure 9).
2) **BiLSTM+DNN fine-tuning**: fine-tuning using a BiLSTM that processes the Transformer's output embedding as a sequence both from left to right and from right to left, subsequently followed by a Deep Feed-Forward network (Figure 8).

Finally, each model leverages a Softmax layer to generate the final prediction among the 18 possible classes. In this way, we could investigate the effects of exposure to financial-related data sets during training, but also the contributions of different Neural Language models such as Autoencoding with different masking approaches (BERT, RoBERTa, and FinBERT), the generalized AutoRegressive (XLNet), and, finally, the Bidirectional Recurrence as a single method (baseline) and in combination with AE and AR language models.

D. FINE-TUNING NETWORKS OPTIMIZATION

Since the models under consideration present a wide heterogeneity in terms of specific hyper-parameters and we aim at a fair comparison among the strategies and language models, we decided to optimize separately each model for each strategy on the validation set. Due to the higher time

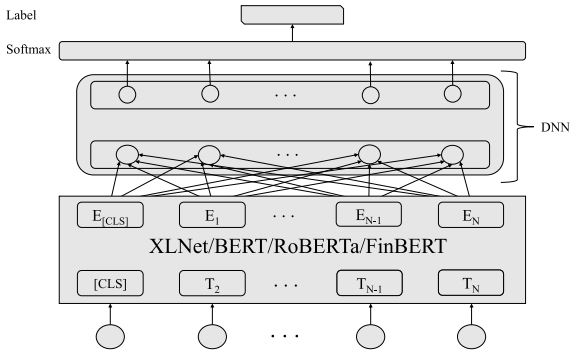


FIGURE 9. Transformer networks with DNN architecture.

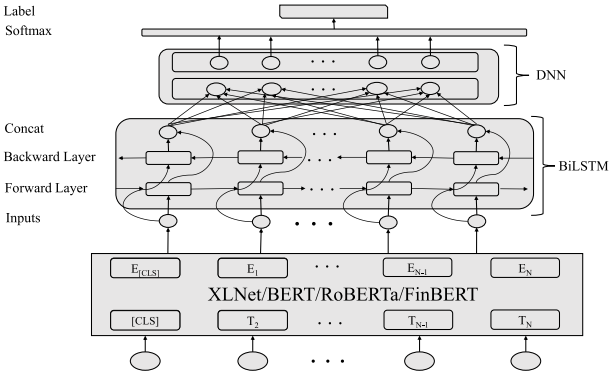


FIGURE 10. Transformer networks with BiLSTM and DNN architecture.

complexity of "large" models and the size of the training and validation sets, there were slight differences in the way we optimized "base" and "large" models.

- For the "base" models, we evaluated 50 different combinations of hyper-parameters when fine-tuned with the DNN architecture
- For "large" models, we evaluated 12 different combinations of hyper-parameters when fine-tuned with the DNN architecture
- For the BiLSTM+DNN models, we evaluated 35 different combinations for each model

Overall, we evaluated 315 models over the same validation sets (balanced and imbalanced). The comparison among all the models is presented in Table 2. In this preliminary analysis, using the validation set, the XLNet-Large (Bi-LSTM+DNN) outperforms all the models for every metric on both the balanced and imbalanced sets.

On the other hand, the two fine-tuning strategies we evaluated achieve very similar performance for AE language models, especially when pre-trained with the NSP task (BERT and FinBERT), but make a significant difference for AR language models where the introduction of bi-directional fine-tuning permits to achieve the best performance and considerably better performance, according to all the metrics, when compared with the same model fine-tuned with the DNN strategy. At the end of this grid-search step, we selected the XLNet-Large+BiLSTM+DNN as the final model for the Document Format Reconstruction task. We named this pre-trained transformer model **SEC-former** and made it publicly available.[4] The optimal parameters of the model are reported in Table 3.

VI. RESULTS

This section presents the results achieved with our final model (SEC-former) for the Document Format Reconstruction task. In particular, we present three groups of results:

1) Results of the experiments performed to assess the capability of the SEC-former of reconstructing 10-K filings in different conditions according to the three test sets presented in Section IV-B.
2) We investigate if SEC-former preserves the document similarity that plays a fundamental role in Portfolio Optimization tasks (See [1], [5]).
3) We provide the qualitative results achieved on a practical use-case where we reconstruct the SEC-like format of the 2022 10-K of an American multinational technology company (Intel Corporation) that was not included in the dataset because of its custom organization of the document.

A. RESULTS ON THE TEST SETS

Using the balanced and imbalanced test sets with documents from 2011 until 2021, we evaluated the ability of the model to generalize on documents from companies that have not been considered when building the training and validation sets (the "Mutual exclusion of companies" setting described in Section IV-B). We used the randomly balanced test set to measure the accuracy over the 18 possible classes and the imbalanced test set to measure the performance with the actual statistical distribution of items.

Furthermore, we investigated the results achieved by SEC-former while reconstructing the format of 765,315 paragraphs from the 2022 financial reports, which had not been used in any design step, providing a real-scenario setting for the evaluation (2022 Test set presented in Section IV-B) without any filtering over the companies, thus allowing the "Copy&Paste" effect highlighted by [5]. Figure 11 shows the results achieved by SEC-former on the three test sets.

According to the results presented in Table 4, SEC-former exhibits similar performance on the imbalanced test set with the mutual exclusion of companies and on the 2022 test set. This last consideration proves that the model achieved item recognition capabilities that are independent of the "Copy&Paste" effect, which minimally affects the performance, only slightly higher on the 2022 test set for all metrics. Table 5 shows the confusion matrix of SEC-Former over the balanced test set. Considering that the test set contains 2771 samples per item, one can observe that most items are most often correctly recognized, especially those belonging to Item 1 - Item 3 and to the last ones (Item 10 - Item 15). The most critical items

[4]https://github.com/sowide/SEC-former

TABLE 3. Parameters and architecture of SEC-former.

BiLSTM's architecture	**Two stacked LSTMs, each having 1024 units, and a dropout of 0.2 between them**
DNN's architecture	**A neural network with two hidden layers of 512 and 256 units, and a dropout of 0.2 after these layers.**
Hidden & Attention dropout	0.1
CLS token dropout	0.0
Batch size	32
Learning rate	5e-5

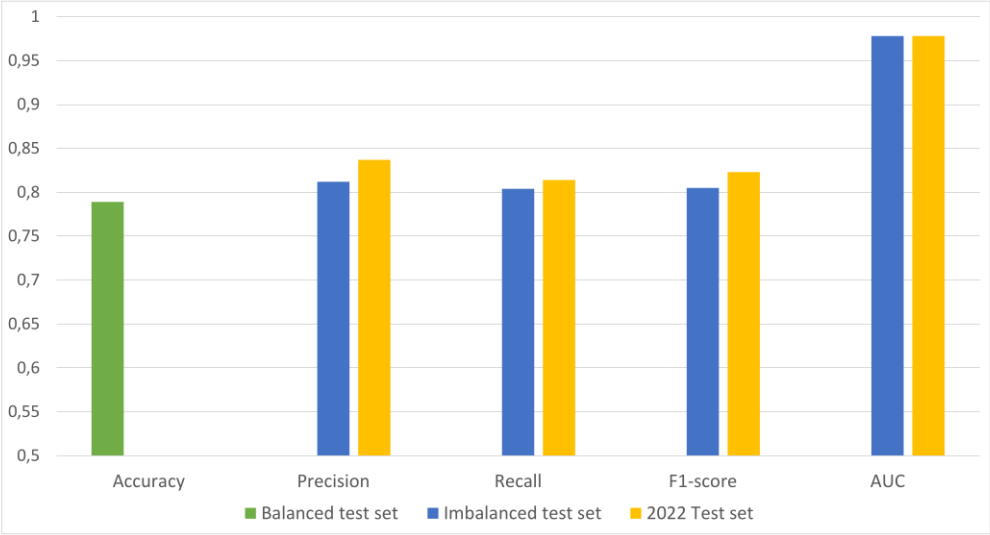


FIGURE 11. Results achieved by the final model on the three test sets (the balanced test set (2011-2021), the imbalanced test set(2011-2021), and the 2022 test set).

TABLE 4. Results of the best model on test sets.

Dataset	Accuracy	Precision	Recall	F1-score	AUC
Balanced test set	0.789	-	-	-	-
Imbalanced test set	-	0.812	0.804	0.805	0.978
Test set 2022	-	0.837	0.814	0.823	0.978

are Item 4 (`Mine Safety Disclosure`) and Item 9 (`Changes in and Disagreements With Accountants`), which are the least represented in the dataset because their content is often very short or absent (See Figure 6). The accuracy on the balanced test set over the 18 classes is equal to 0.789.

B. DOCUMENT SIMILARITY

Measuring document similarity among consecutive 10-K reports is crucial for investors due to the relationships identified by [5] among changes in the document and abnormal future returns. In light of this, the Document Format Reconstruction task plays a key role in enabling the analysis and measure of similarity at the items level.

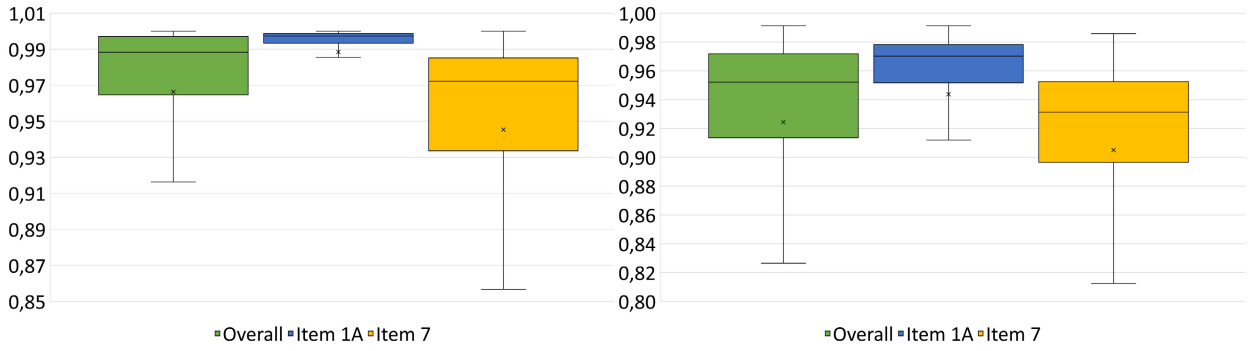
For this reason, it is not sufficient to measure the performance of SEC-former only in terms of the number of single paragraphs correctly classified. It is primarily important that the resulting overall item obtained by applying SEC-former offers the chance of measuring a trustworthy document similarity, despite possible errors introduced by the classifier. To achieve such a goal, we considered all paragraphs in the **2022 Test set**, focusing on the longest and over-represented in the dataset *Item 1A* and *Item 7* to measure the document similarity between the original clean item and the version reconstructed by SEC-former. Indeed, being the most frequently represented ones, these two items largely affect the overall Document similarity.

1) For each filing in 2022, we labeled each paragraph of the two items according to the parser introduced in Section IV-C
2) We reconstructed the same items with SEC-former
3) We computed the document similarity between the real and the reconstructed one with the same methodology used to analyze the document similarity in Lazy Prices (Tf-Idf encoding) and the Cosine-similarity with Doc2Vec embeddings ([8]).
4) Finally, we repeated the analysis on the entire original document and the reconstructed one.

The results of this experiment are shown in Figures 12a and 12b. As one can see, despite some possibly misclassified paragraphs that are therefore missing, the

TABLE 5. Confusion matrix of SEC-Former over the balanced test set.

	Item 1	Item 1A	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 7A	Item 8	Item 9	Item 9A	Item 9B	Item 10	Item 11	Item 12	Item 13	Item 15
Item 1	**2321**	151	24	6	21	8	65	129	6	14	0	2	3	13	2	0	6	0
Item 1A	268	**2371**	4	5	4	10	28	43	4	3	3	4	6	3	1	3	11	0
Item 2	90	16	**2572**	9	2	1	29	16	11	1	2	1	10	0	0	0	9	2
Item 3	11	25	11	**2481**	138	4	10	14	4	18	19	3	16	4	1	1	8	3
Item 4	3	4	1	184	**1975**	62	206	11	30	0	55	0	7	200	4	13	16	0
Item 5	11	44	6	4	10	**2428**	107	27	6	5	3	0	11	0	13	76	8	12
Item 6	101	25	27	7	13	48	**2091**	334	28	26	6	6	6	1	13	0	36	3
Item 7	170	75	14	8	2	20	166	**2119**	51	114	7	5	6	0	1	0	6	7
Item 7A	16	43	1	1	23	10	213	246	**1922**	101	37	39	20	0	2	6	2	89
Item 8	19	6	2	35	1	15	62	261	22	**2121**	12	33	18	0	12	0	11	141
Item 9	25	50	14	103	32	29	186	151	321	308	**1077**	130	178	9	6	9	41	102
Item 9A	6	5	2	16	0	4	26	60	38	285	50	**2140**	22	0	1	0	2	114
Item 9B	10	8	41	64	65	4	23	33	31	90	148	103	**1884**	145	41	4	38	39
Item 10	5	6	0	5	287	1	3	0	0	1	0	1	45	**2324**	41	2	45	5
Item 11	5	0	1	0	1	16	14	12	0	1	0	2	19	10	**2624**	42	23	1
Item 12	1	3	0	0	0	26	1	2	0	4	2	1	0	2	109	**2604**	15	1
Item 13	26	27	12	9	3	41	41	38	10	54	13	6	14	41	58	44	**2128**	206
Item 15	11	3	1	11	0	6	19	58	12	427	9	21	10	0	3	0	5	**2175**



(a) Box plot of the Tf-Idf embedding results.

(b) Box plot of the Doc2Vec embedding results.

FIGURE 12. Document similarity results with Tf-Idf embeddings (a) and Doc2Vec embeddings (b).

reconstructed items and documents are very similar to the original ones. The analysis with the Bag-Of-Words model with Tf-Idf proves that the overlap in terms of words is almost total, with an average similarity equal to 0.97. The analysis with Doc2Vec embeddings shows that the original and the reconstructed documents are also semantically similar without considering the paragraphs' or words' order, with an average similarity equal to 0.93. In light of this, we can conclude that the documents reconstructed by the SEC-former can be effectively used for downstream tasks based on document similarities, such as Portfolio Optimization and others.

C. PRACTICAL USE-CASE

Regardless of the quantitative results achieved, it is interesting to verify how SEC-former works on a practical use case in which we ask SEC-former to reconstruct a complex document that exhibits a very different structure from the one suggested by SEC.

We selected *Intel Corporation*, an American multinational corporation and technology company headquartered in California. We chose this company because, starting in 2021, Intel adopted a new structure for its filings,[5] which differs from the one SEC recommends, although the company makes a cross-reference index available to show readers the possible correspondence with the SEC-like structure reported in Table 6.

TABLE 6. Cross-reference index provided by Intel in its 2021 filing.

Item number	Pages	Item number	Pages
Item 1	2-36, 48-49, 66, 82-85	Item 8	68-113
Item 1A	50-63	Item 9	Not Applicable
Item 1B	Not Applicable	Item 9A	114
Item 2	12, 64	Item 9B	67
Item 3	105-110	Item 10	Ref. Proxy Statement
Item 4	Not Applicable	Item 11	Ref. Proxy Statement
Item 5	9, 64-65	Item 12	Ref. Proxy Statement
Item 6	Reserved	Item 13	Ref. Proxy Statement
Item 7	18-47, 77-82	Item 14	Ref. Proxy Statement
Item 7A	49	Item 15	115-119

This task yielded some very interesting results. SEC-former did not find any paragraph belonging to *Item 10*, *Item 11*, *Item 12*, *Item 13*, *Item 14*, which is correct because, as explained in the cross-reference index provided by Intel, these items are incorporated by reference in another document (2022 Proxy Statement). Also, no paragraphs belonging to *Item 15* were detected, which is also correct because this item is composed of images and tables, which have been removed from the document during the parsing stage. We reported some extracts of the reconstructed items in Table 7, to show that each item's content conforms SEC

[5]Intel filings are available on the SEC website, at the following link

TABLE 7. Extracts of certain items reconstructed from the Intel 2022 10-K by SEC-former model.

Content	Item
...Our investments in new businesses, products, and technologies are inherently risky and do not always succeed.... ...We are subject to the risks of product defects, errata, or other product issues... ...We face risks related to security vulnerabilities in our products... ...We are subject to risks associated with litigation and regulatory matters... ...We face risks related to sales through distributors and other third parties...	1A
As of December 25, 2021, our major facilities (Square Feet in Millions) consisted of: Owned facilities 31, 24, 55; Leased facilities 1, 5, 6; Total facilities 32, 29, 61 (United States, Other Countries, Total). Our principal executive offices are located in the US. For more information on our wafer fabrication and our assembly and test facilities, see "Manufacturing Capital" within Fundamentals of Our Business. The facilities described above are suitable for our present purposes, and the productive capacity in our facilities is being utilized or being prepared for utilization as we continue to make investments to expand our manufacturing capacity. We do not identify or allocate assets by operating segment, as they are interchangeable in nature and used by multiple operating segments. For information on net property, plant and equipment by country, see "Note 6: Other Financial Statement Details" within the Financial Statements and Supplemental Details.	2
We are regularly party to various ongoing claims, litigation, and other proceedings, including those noted in this section. In the first quarter of 2021, we accrued a charge of $2.2 billion related to litigation involving VLSI, described below. Excluding this charge, management at present believes that the ultimate outcome of these proceedings, individually and in the aggregate, will not materially harm our financial position, results of operations, cash flows, or overall trends; however, legal proceedings and related government investigations are subject to inherent uncertainties, and unfavorable rulings, excessive verdicts, or other events could occur... ...We have had IP infringement lawsuits filed against us, including but not limited to those discussed below. Most involve claims that certain of our products, services, and technologies infringe others' IP rights. Adverse results in these lawsuits may include awards of substantial fines and penalties, costly royalty or licensing agreements, or orders preventing us from offering certain features, functionalities, products, or services...	3
Operating income increased $37 million, driven by higher revenue due to recovery in the embedded and communications market segments from COVID-19 lows, partially offset by a decrease in the cloud market segment... ...Our total revenue grew from $62.8 billion in 2017 to $79.0 billion in 2021, representing 6% CAGR. In 2021, revenue was $79.0 billion, up $1.2 billion, or 1%, from 2020. CCG revenue grew 1% due to continued strength in notebook demand and recovery in desktop demand, partially offset by lower notebook ASPs due to strength in the consumer and education market segments. CCG adjacent revenue decreased primarily due to the continued ramp down from the exit of our 5G smartphone modem and Home Gateway Platform businesses... ...Our effective tax rate decreased in 2021 compared to 2020, primarily driven by one-time tax benefits due to the restructuring of certain non-US subsidiaries as well as a higher proportion of our income in non-US jurisdictions. As a result of the restructuring, we established deferred tax assets and released the valuation allowances of certain foreign deferred tax assets... ...Financing cash flows consist primarily of payment of dividends to stockholders, issuance and repayment of short-term and long-term debt, repurchases of common stock, and proceeds from the sale of shares of common stock through employee equity incentive plans...	7
We have established currency risk management programs to protect against currency exchange rate risks associated with non-US dollar forecasted future cash flows and existing non-US dollar monetary assets and liabilities. We may also hedge currency risk arising from funding of foreign currency-denominated future investments. We may utilize foreign currency contracts, such as currency forwards or option contracts in these hedging programs... ...We are exposed to interest rate risk related to our fixed-rate investment portfolio and outstanding debt. The primary objective of our investment policy is to preserve principal and provide financial flexibility to fund our business while maximizing yields, which generally track the US dollar three-month libor... ...We are exposed to equity market risk through our investments in marketable equity securities, which we typically do not attempt to reduce or eliminate through hedging activities...	7A
We have audited the accompanying Consolidated Balance Sheets of Intel Corporation (the Company) as of December 25, 2021 and December 26, 2020, the related Consolidated Statements of Income, Comprehensive Income, Cash Flows and Stockholders' Equity for each of the three years in the period ended December 25, 2021, and the related notes (collectively referred to as the "Consolidated Financial Statements")...	8

recommendations in terms of topic, as shown in Table (1). These qualitative results of the reconstructed 10-K filing confirm that the SEC-former is able to learn contextual information from paragraphs.

VII. CONCLUSION

We investigated Recurrence-based, Autoregressive, and Autoencoding-based language models to perform the Document Format Reconstruction task over the 10-K SEC filings, also proposing a bidirectional fine-tuning procedure that exhibits better performance with respect to Feed-forward approaches. Finally, we made the pre-trained model resulting from our research publicly available to the scientific community for further analysis.

Future developments are related to the possible improvements of the SEC-former that can be achieved by introducing logic constraints on the order of the paragraphs when performing the reconstruction. We also plan to investigate the behavior of this model in a zero and few-shoot learning setting when applied to other financial reports, such as Quarterly reports (10-Q) and Earning calls.

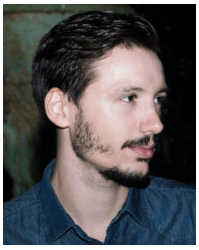
REFERENCES

[1] G. Adosoglou, S. Park, G. Lombardo, S. Cagnoni, and P. M. Pardalos, "Lazy network: A word embedding-based temporal financial network to avoid economic shocks in asset pricing models," *Complexity*, vol. 2022, pp. 1–12, Apr. 2022.
[2] A. Khadjeh Nassirtoussi, S. Aghabozorgi, T. Ying Wah, and D. C. L. Ngo, "Text mining of news-headlines for FOREX market prediction: A multi-layer dimension reduction algorithm with semantics and sentiment," *Exp. Syst. Appl.*, vol. 42, no. 1, pp. 306–324, Jan. 2015.
[3] M. Van de Kauter, D. Breesch, and V. Hoste, "Fine-grained analysis of explicit and implicit sentiment in financial news articles," *Exp. Syst. Appl.*, vol. 42, no. 11, pp. 4999–5010, Jul. 2015.
[4] C. Zopounidis, M. Doumpos, and P. M. Pardalos, *Handbook of Financial Engineering*. Berlin, Germany: Springer, 2010.
[5] L. Cohen, C. Malloy, and Q. Nguyen, "Lazy prices," *J. Finance*, vol. 75, no. 3, pp. 1371–1415, Jun. 2020.
[6] T. Laughran and B. Mcdonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," *J. Finance*, vol. 66, no. 1, pp. 35–65, 2011.
[7] T. Dyer, M. Lang, and L. Stice-Lawrence, "The evolution of 10-K textual disclosure: Evidence from latent Dirichlet allocation," *J. Accounting Econ.*, vol. 64, nos. 2–3, pp. 221–245, Nov. 2017.
[8] G. Adosoglou, G. Lombardo, and P. M. Pardalos, "Neural network embeddings on corporate annual filings for portfolio selection," *Exp. Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 114053.
[9] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Doc2Vec," in *Proc. 24th Int. Conf. World Wide Web*, vol. 32, 2015, pp. 29–30.
[10] P. Cerchiello, G. Nicola, S. Rönnqvist, and P. Sarlin, "Assessing banks' distress using news and regular financial data," *Frontiers Artif. Intell.*, vol. 5, 2022, Art. no. 871863.
[11] F. Mai, S. Tian, C. Lee, and L. Ma, "Deep learning models for bankruptcy prediction using textual disclosures," *Eur. J. Oper. Res.*, vol. 274, no. 2, pp. 743–758, Apr. 2019.
[12] A. Glodd and D. Hristova, "Extraction of forward-looking financial information for stock price prediction from annual reports using NLP techniques," Ph.D. thesis, Univ. Haway Manoa, 2023.
[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.
[14] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[15] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100, 000+ questions for machine comprehension of text," 2016, *arXiv:1606.05250*.
[16] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," 2018, *arXiv:1804.07461*.
[17] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," 2015, *arXiv:1506.06724*.
[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[19] F. Hamborg, N. Meuschke, C. Breitinger, and B. Gipp, "News-please: A generic news crawler and extractor," in *Proc. 15th Int. Symp. Inf. Sci.*, Mar. 2017, pp. 218–223.
[20] A. Gokaslan and V. Cohen. (2019). *Openwebtext Corpus*. [Online]. Available: http://Skylion007.github.io/OpenWebTextCorpus
[21] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," 2019, *arXiv:1906.08237*.

[22] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," 2019, *arXiv:1901.02860*.
[23] B. Uria, Ma.-A. Cote, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation," 2016, *arXiv:1605.02226*.
[24] A. Kurniasih and L. P. Manik, "On the role of text preprocessing in BERT embedding-based DNNs for classifying informal texts," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, 2022.
[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," 2016, *arXiv:1603.02754*.
[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[27] Y. Zhao, G. Chetty, and D. Tran, "Deep learning with XGBoost for real estate appraisal," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 1396–1401.
[28] M. Jiang, J. Liu, L. Zhang, and C. Liu, "An improved stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms," *Phys. A, Stat. Mech. Appl.*, vol. 541, Mar. 2020, Art. no. 122272.
[29] G. Lombardo, M. Pellegrino, G. Adosoglou, S. Cagnoni, P. M. Pardalos, and A. Poggi, "Machine learning for bankruptcy prediction in the American stock market: Dataset and benchmarks," *Future Internet*, vol. 14, no. 8, p. 244, Aug. 2022.
[30] D. Araci, "FinBERT: Financial sentiment analysis with pre-trained language models," 2019, *arXiv:1908.10063*.

GIANFRANCO LOMBARDO received the Ph.D. degree from the University of Parma, in 2021. In Fall 2019, he was a Visiting Researcher with the Center for Applied Optimization, Herbert Wertheim College of Engineering, University of Florida. He is currently an Assistant Professor with the Department of Engineering and Architecture, University of Parma. His research interests include deep learning and natural language processing for the financial domain.

GIUSEPPE TRIMIGNO received the B.Eng. degree in computer engineering from the University of Parma, in 2022, where he is currently pursuing the M.Sc. degree in computer engineering (AI curriculum). His main research interests include deep learning and natural language processing.



MATTIA PELLEGRINO received the M.Eng. degree in computer engineering from the University of Parma, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include deep learning and reinforcement learning.

STEFANO CAGNONI (Senior Member, IEEE) received the degree in electronic engineering from the University of Florence, in 1988, where he is currently pursuing the Ph.D. degree. In 1994, he was a Visiting Scientist with the Whitaker College Biomedical Imaging and Computation Laboratory, Massachusetts Institute of Technology. He was a Postdoctoral Researcher with the University of Florence, until 1997. Since 1997, he has been with the University of Parma, Parma, Italy, where he has been an Associate Professor, since 2004. He has published more than 150 papers in international journals and conference proceedings. His research interests include artificial intelligence, with particular regard to evolutionary computation and its applications to image analysis and pattern recognition.

Open Access funding provided by 'Università degli Studi di Parma' within the CRUI CARE Agreement