Dig Here! Extracting and Using Knowledge from Financial Audit Reports

Sachin Pawar, Manoj Apte, Aditi Pawde, Sushodhan Vaishampayan

Girish Keshav Palshikar, Akshada Shinde

TCS Research, Tata Consultancy Services Ltd., Pune, India - 411013
{sachin7.p, manoj.apte, pawde.aditi, sushodhan.sv, gk.palshikar, akshada.shinde}@tcs.com

Abstract

Financial audits establish trust in the governance and processes in an organization, but they are time-consuming and knowledge intensive. To increase the effectiveness of financial audit, we address the task of generating audit suggestions that can help auditors to focus their investigations. Specifically, we present NLP techniques to extract hidden knowledge from a corpus of past financial audit reports of many companies, and use it for generating audit suggestions. The extracted knowledge consists of a set of automatically identified sentences containing adverse remarks, the financial variables mentioned in each sentence and automatically assigned XBRL categories for them, since XBRL is a standardized taxonomy in the financial domain. In the absence of suitable labeled data, we adopted a weak supervision approach. We designed a set of high precision linguistic rules to identify adverse remark sentences, created automatically labeled training data using them, and trained BERT-based and other classifiers to identify such sentences. We next presented novel techniques (which are either unsupervised or zero-shot) to assign zero, one, or more XBRL categories to any given adverse remark sentence. We evaluated the proposed approaches, on a large corpus of real-life financial statements and audit reports, against competent baselines. Given a company's financial statements (already identified as suspicious), and given a subset of financial variables in them that contribute to suspiciousness, we match these with the extracted knowledge base and identify aligned adverse remarks that help the auditor in focusing on specific directions for further investigations.

Introduction

Financial audit is a complex and knowledge-intensive discipline within Accounting for trust-building and ensuring quality of governance in an organization (Arens, Elder, and Beasley 2016), (Nigrini 2020). It validates internal controls, safeguards the assets, evaluates current and future risks, and provides suggestions for improving governance, ensures processes are followed as required, ensures compliance with standards, guidelines, laws and regulations and ensures that the reported financial information is fair and accurate. Stakeholders use audited *financial statements* (FS), such as balance sheet, income statement, cash-flow statement etc., for decision-making. Examples: regulatory bodies use FS to check compliance, tax departments use them to validate taxes paid and benefits claimed, investors use them to estimate the financial health of the company etc.

A *financial auditor* is responsible for carrying out financial audit, and in particular for validating and certifying that the financial data mentioned in the FS of a company is fair and accurate. This validation is typically done by collecting evidence from (a) trails of business processes followed in the company (e.g., payment receipts, transaction statements, contractual documents, letters to and from banks, authorities, customers and suppliers etc.); as well as from (b) personal inspections (e.g., of warehouses). If the data in FS is consistent with the evidence collected, then the auditor declares in an *audit report* that the FS are free from material misstatement, fair and accurate and presented in accordance with the relevant accounting standards. If not, the auditor makes *adverse remarks* about the detected or potential instances of non-conformance, misinformation, irregularities, inconsistencies, errors, inaccuracies, frauds, lapses, non-compliance, violations etc. Adverse remarks often also include auditors' suggestions for improvement. Clearly, knowledge and experience of the auditor play a vital role to efficiently and effectively carry out an audit.

Considering these challenges, our aim is to assist the auditors through an *intelligent audit assistance system* (Figure 1). Suppose we have a historical corpus $\mathbf{D}$ of past FS and associated audit reports for several companies for different years. Suppose also that we are given a FS B for a particular company for a particular year. We assume B is already known to be *anomalous*, and we also assume that we already know an *explanation* of why B might be anomalous. The task handled in this paper is to identify relevant *audit suggestions* from $\mathbf{D}$ that the auditor can use to guide the progress of the audit process i.e., to explore whether or not the suggested explanations are correct.

Machine Learning (ML) techniques have been used to detect misinformation in FS (see (Ashtiani and Raahemi 2021) for a review). While mostly supervised ML techniques are used (e.g., (Kirkos, Spathis, and Manolopoulos 2007)(Perols 2011)), *unsupervised* techniques are used to determine whether B (represented as a feature vector of *financial variables* (FV)) is *anomalous* with respect to $\mathbf{D}$ (Shinde et al. 2022). Informally, a FV refers to something that has a monetary value; each FS is essentially a set of (FV-name, value) pairs. If B is detected as anomalous, we then use an explana-

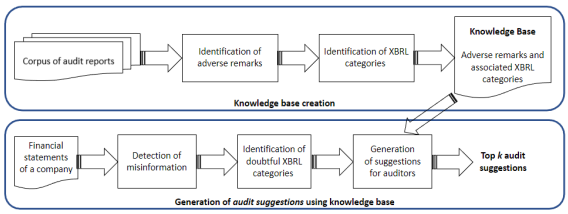

Figure 1: Intelligent audit assistance system.

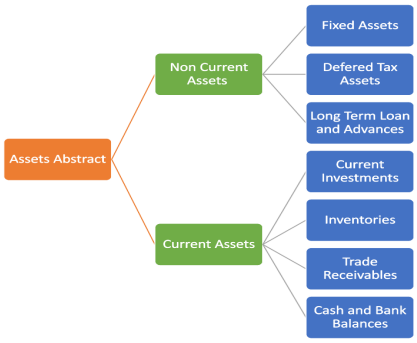

Figure 2: A snapshot of XBRL category hierarchy.

tion generation technique (e.g., (Vaishampayan et al. 2022)) to identify reasons why B is anomalous with respect to $\mathbf{D}$; in particular, we identify a subset V_B of FV in B as the explanation of why B is anomalous. In this paper, we address the task of mining $\mathbf{D}$ for providing additional actionable suggestions to the auditor, as to what *specifically* might be wrong with the data in B for V_B. Basically, we identify all adverse remarks in $\mathbf{D}$ related to each FV in V_B, and suitably pick one or more of these to generate actionable suggestions for the auditor. With this help, auditors can focus on collecting evidence to validate the detected anomalies and explanations generated for them by ML techniques. This will help in reducing efforts and bias in the audit process.

The names and mentions of FV in FS, reports and audit reports are influenced by norms, practices and standards within industries or nations. To facilitate comparison of FS across companies and across years, a taxonomy of standardized classes of FV (called *categories*) is defined in the *eXtensible Business Reporting Language (XBRL)* standard[1]. In this paper, we *flatten* the XBRL 2010 taxonomy[2], and use the resulting 1724 categories ignoring the hierarchical relationships among them (Fig. 2). We ignore categories that refer to "Other" kind of financial information, so the number of remaining categories we consider is 1437.

In this paper, we focus on automatically extracting a *knowledge-base* (KB) of actionable knowledge elements from a given corpus AR of past audit reports. Each knowledge element consists of the tuple (S, v, X_v), where S is a an adverse remark sentence, v is the text fragment in S which is a mention of a FV in S, X_v is a set of XBRL categories corresponding to v. In this paper, we develop techniques for (i) **Task A:** Identifying all adverse remark sentences from AR; and (ii) **Task B:** Assigning 0, 1 or more XBRL categories to each adverse remark sentence S, along with identifying mentions (if any) of them in S. Usually, an XBRL category will match with a mention of a FV in S. We also demonstrate how this KB could be used to generate *audit suggestions* for a given FS B (already detected as anomalous) and a given set of FV in B (already detected as making B anomalous); Table 1. Each sentence there can be considered as a potential audit suggestion; e.g., if V_B includes `Long Term Loans and Advances` as a FV that makes given B anomalous, then sentence (4) can be mapped to a useful audit suggestion (*"check if the company has defaulted on loan payments"*), identifiable because this FV and this sentence are both mapped to XBRL category Current Liabilities.

Task A (modeled as a sentence classification problem) is challenging, because adverse remarks are expressed in a large variety of linguistic ways. Also, presence of negation complicates identification of adverse remarks. Identifying XBRL categories that *match* with a given sentence (Task B) is challenging because a FV is often expressed very differently in text than the standard name of an associated XBRL category. For instance, XBRL category Trade Receivables should match with mentions like `recovery of dues, charges recoverable` and XBRL category Inventories should match with mentions like `warehouse stock, inventory holdings, unsold goods, unused material`. Names of XBRL categories often overlap (e.g., taxes payable wealth tax, wealth tax provision, wealth tax receivable) and identifying the correct one among them is tricky. Our techniques can also be used to match XBRL categories in sentences which are *not* adverse remarks; e.g., **`Hire purchase loans`** `are repayable within one year and shown under` **`Other Current Liabilities`**. Some adverse remark may not be associated with *any* XBRL category; e.g., `Continuing failure to correct major weakness in the internal control system was noticed.`. To our knowledge, this is the first attempt to generate audit suggestions for auditors using XBRL categories and adverse remarks extracted from past audit reports.

Related Work

The problem of identifying adverse remark sentences from a corpus of audit reports is a task of sentence classification. FinBERT has been used in several applications; e.g., (Yadav et al. 2019) used it to perform sentiment analysis on financial news to understand effect of public sentiments on the stock price of a company. These models classify a sentence as having positive, negative or neutral sentiment. However, a negative sentiment does not always imply adverse remark; e.g. `company has incurred loss of Rs. 320 lacs` is not an adverse remark (it states a financial fact) though it has a negative sentiment.

Task B is similar to Extreme Multi-Label Classification (XMLC), since there are 1437 XBRL categories and a single sentence can contain multiple XBRL categories. Super-

[1] www.xbrl.org

[2] `https://www.mca.gov.in/XBRL/pdf/Revised_taxonomy.zip`

•During the course of audit we also noticed that YY bank has issued notice under SARFAESI ACT, on **factory building**$_{\text{Factory Building}}$ and **current assets**$_{\text{Current Assets}}$ of the company.
•The company is carrying **P & P Expenses** and **issue related expenses**$_{\text{Share Issue Expenses}}$ of Rs. YY as **other current assets**$_{\text{Current Assets Other}}$ which in our opinion needs to be write off.
•The Company has not deposited amounts of Rs. YY lacs to the **Investor Education & Protection Fund** against **unpaid fixed deposit liabilities including interest thereon**$_{\text{Current Liabilities}}$.
•The Company has defaulted in payment of **bank loans**$_{\text{Current Liabilities}}$ during the year preceding the previous financial period and continued in this financial year.

Table 1: Adverse remarks, mentions of financial variables (bold) in them and associated XBRL categories (subscript).

vised ML approaches are difficult to use due to large time and efforts needed from expert accountants in creating labeled training data. SurveyCoder (Patil and Ravindran 2015) is one unsupervised approach wherein it assigns one or more codes from the given code-frame to customer response for an open-ended question. It checks for overlap of semantic unit-based representation of a code with each sentence in a response. (Devine and Blincoe 2022) is an unsupervised approach for XMLC which tags posts on tech forums like StackOverflow with relevant categories. Embeddings of the post and category tags obtained using deep Transformer based models are compared using cosine similarity to obtain list of matching tags. Tags having similarity above a predefined threshold are reported as the recommended tags. (Loukas et al. 2022) models XBRL tagging as named entity recognition. However, it uses a supervised approach and considers only 139 XBRL categories, whereas we consider all 1437 XBRL categories to develop an interpretable and unsupervised method.

Adverse Remark Classification (Task A)

We use a sentence classification approach to predict whether a sentence contains any *adverse_remark* or not (*NA*). Due to unavailability of a public labeled dataset for adverse remarks in audit reports, we propose a weakly supervised approach which works in two steps. In step 1, a set of high-precision linguistic rules are used to create a labeled dataset. In step 2, a BERT-based sentence classifier is trained using the labeled data created in the first step. A motivation to train a classifier is to improve recall by learning a more generalized model of the high-precision linguistic rules. The proposed technique does not need manual annotation for training, making it more scalable.

Proposed linguistic rules are fairly intuitive and are divided into several categories (Table 2). R_1 labels a sentence as *adverse_remark* if it contains a verb with negative sentiment from audit perspective and which is not modified by a negation indicating word (e.g., `no`, `neither`) in the dependency tree. R_2 identifies a sentence as *adverse_remark* if it contains a verb with positive sentiment but which is modified by a negation indicating word. For R_2, the verb should also have its object (*dobj* or *nsubjpass*) related to key financial concepts such as `tax`, `debt`, `dues` etc. R_3 and R_4 (and R_5 and R_6) are similar rules which check for presence of certain nouns (respectively, adjectives) instead of verbs. In addition to labelling a sentence as *adverse_remark*, the rules also assign *NA* label (i.e., *not adverse_remark*) if negation is present in case of R_1, R_3 and R_5 and if negation is not present in case of R_2, R_4 and R_6. R_7 marks a sentence as *adverse_remark* if certain negative opinion markers are present. R_8 labels a sentence as *adverse_remark* if it contains the word `except` followed by a clause containing some negative word without any negation (`no`, `not`, etc.). Thus on some sentences the rules would not predict *any* label.

Given the training data containing sentences from a corpus labelled using linguistic rules, we designed a sentence classifier based on BERT that uses additional attention layers and keyword-based features for obtaining better sentence representations (we denote this classifier by M_{BERT}). An input sentence S is first passed through the pre-trained BERT model to obtain (i) [CLS] token encoding which provides the representation of the entire input text S, and (ii) the representations for each word in S.

$$\mathbf{x_{CLS}}, X = BERT(S) \tag{1}$$

Here, $\mathbf{x_{CLS}} \in \mathbb{R}^{768}$ and $X \in \mathbb{R}^{L \times 768}$ where L[3] is the maximum number of words in any input sentence. Let $X_i \in \mathbb{R}^{768}$ be the representation for the i^{th} word in S. We use an attention layer similar to the one in (Basiri et al. 2021) so that the contribution of each word in S is determined based on its importance: $a_i = \mathbf{w_a}^T \cdot X_i + b$. Here, $\mathbf{w_a} \in \mathbb{R}^{768}$ and $b \in \mathbb{R}$ are the weight vector and bias of the attention layer, respectively. $a_i \in \mathbb{R}$ is the score for the i^{th} word as computed by the attention layer. These scores are normalized across all the words in S to obtain final attention weights which are used to obtain a weighted average of word representations.

$$\alpha_i = \frac{exp(a_i)}{\sum_{j=1}^{L} exp(a_j)}; \mathbf{x_w} = \sum_{i=1}^{L} \alpha_i \cdot X_i \tag{2}$$

In addition to the text representation by the pre-trained BERT model, we also try to capture presence of important *keywords* (Table 2) that may indicate presence of an *adverse_remark*. The presence or absence of each of the K *keywords* in S is represented by a binary vector $\mathbf{x_{kw}} \in \mathbb{R}^K$. Finally, the overall representation of the input sentence is obtained by concatenating it with the representations obtained in Equations 1 and 2 as: $\mathbf{x_{final}} = [\mathbf{x_{CLS}}; \mathbf{x_w}; \mathbf{x_{kw}}]$. This final representation is then passed through two linear transformation layers to obtain the final output which is a probability distribution over two labels – *adverse_remark* and *NA*. The model is then trained to minimize the cross entropy loss.

We used the following baselines for this classification task:

RF, LR, XGBoost, SVM_RBF: We train Random Forest, Logistic Regression, Support Vector Machine with RBF Kernel and XGBoost classifiers on the training dataset D_{TR}

[3]We use $L = 128$ in our experiments.

Rule	Example verbs/nouns in Rule's Gazette	Example
R_1: Negative verbs without negation	overstate, understate, default, inflate, cheat, neglect, hide, fail, miss, ...	The Company has **defaulted** in repayment of dues to the following banks...
R_2: Positive verbs with negation	pay, repay, deposit, provide, receive, charge, ...	Undisputed statutory dues have **not** been regularly **deposited** with authorities.
R_3: Negative nouns without negation	overstatement, understatement, cheating, negligence, fraud, crime, criminal, delay, ...	The amount of alleged **fraud** according to management is Rs. 7,10,31,008/-...
R_4: Positive nouns with negation	payment, repayment, provision, settlement, ...	The company has **not** made any **provision** for depreciation in respect of...
R_5: Negative adjectives without negation	irregular, diverted, undisclosed, unexplained, inoperational, quantifiable, ...	Utilization of GDR proceeds for **undisclosed** purposes indicate violations of the FEMA Act
R_6: Positive adjectives with negation	regular, clear, sufficient, disclosed, ascertained, ...	The Company has **not** been **regular** in depositing undisputed statutory dues.
R_7: Opinion markers	should have been, yet to, failed to, ...	Trade discount **should have been** netted off from Sales.
R_8: Thwarting	overstate, understate, default, inflate, cheat, neglect, hide, delay...	Company was regular in depositing statutory dues, **except** for TDS where there are **delays**.

Table 2: Linguistic rules for creating labeled data

created by our linguistic rules.

FinBERT: We use the pre-trained FinBERT (Araci 2019) model as a classifier.

Ensemble: We create an ensemble of the individual classifiers to improve the classification accuracy as follows. As the linguistic rules are high-precision, we consider predictions of the rules as the final prediction for the sentences where rules are applied (for both *adverse_remark* and *NA* classes). For the remaining sentences, we check the predictions by other classifiers. We consider M_{BERT} predictions as final predictions for remaining sentences except for the following two exception rules. If M_{BERT} predicts *NA* for a sentence but all other classifiers (RF, LR, SVM_RBF and XGBoost) predict *adverse_remark* then we change its final prediction to *adverse_remark*. If M_{BERT} predicts *adverse_remark* for a sentence but at least two other classifiers do not predict *adverse_remark* then we change its final prediction to *NA*.

Assign XBRL Categories to Sentences (Task B)

We explored several unsupervised techniques for this task.

SBfull: This baseline technique, based on (Devine and Blincoe 2022), computes cosine similarity between the Sentence-BERT (Reimers and Gurevych 2019) based embeddings of an XBRL category X and the input sentence S and assigns categories having similarity with S above a threshold θ_0 (we used $\theta_0 = 0.5$).

JS: This simple, unsupervised and interpretable technique uses *Jaccard similarity* between two sets. For each XBRL category X having $|X| = k$ words in it, we consider windows of sizes $k-1$, k, $k+1$ and $k+2$ and obtain all possible contiguous word subsequences in a given sentence S by sliding these windows. The similarity of the category X with the sentence is the highest Jaccard similarity with any of these subsets. While computing similarity, synonyms and morphological variations of a word are taken into consideration and stopwords are ignored. A category is matched with a sentence if the Jaccard similarity thus computed is above a threshold θ_1 (we used $\theta_1 = 0.6$).

SBwin: This technique is similar to **JS**, except that instead of Jaccard similarity, it uses cosine similarity between the Sentence-BERT embeddings of the subsequence (window) of the input sentence and that of the XBRL category (we used threshold $\theta_2 = 0.8$). To improve performance, categories which match at lower similarity threshold ($= 0.3$) with *S* in SBfull are only considered in SBwin.

BM25: Okapi BM25 (Robertson and Zaragoza 2009) is a probabilistic retrieval framework based ranking function that estimates the relevance of documents given a query. We use a sentence S as a query and all XBRL categories as documents. All sentences and XBRL category names are preprocessed to remove stopwords and only lemmatized distinct words from sentences and queries are retained. Relevance score thus obtained is divided by length of the sentence to remove bias due to long sentences. Since there are thousands of XBRL categories, for better efficiency, we shortlist only those that have more than 50% overlap with the given sentence. We consider top m scores for a sentence and retrieve all categories that have relevance score equal to any of these top m scores. (we used $m = 10$).

0SHOT-TC: *Zero-shot text classification (0SHOT-TC)* (Yin, Hay, and Roth 2019) using Natural Language Inference (NLI) assigns a label to a text fragment even without its context such as domain, topic etc. Given a sentence *S* as premise, the hypothesis `This is about [category]` is tested for all XBRL categories using pre-trained BART-large-MNLI model. Categories having entailment score above threshold θ_3 are assigned to *S* (we used $\theta_3 = 0.6$).

Experimental Analysis

Performance on Task A

Input Dataset: We used the web-scraped audit reports made available by authors of (Maka, Pazhanirajan, and Mallapur 2020). We prepared two datasets D_{14}, D_{15} for year 2014 (#audit reports: 3759, #sentences: 325486) and 2015 (#audit reports: 3343, #sentences: 241482). The number of *clean* sentences, after removing very short sentences, noisy sentences such as table rows, and duplicate sentences, is D_{14}:106610 and D_{15}:71019. We used the linguistic rules on

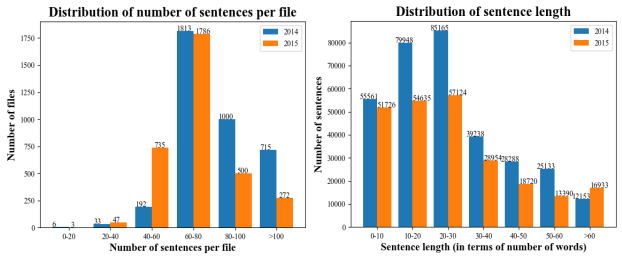

Figure 3: Distributions of no. of sentences per file and sentence length

D_{14} to prepare the training set for our sentence classifier. D_{15} was used as an unseen dataset for evaluating the performance of all methods. Summary statistics for number of words per sentence (see Fig. 3):
D_{14}: average=25.6, stdev=17.1, Q1=13, median:22, Q3=35;
D_{15}: average=26.6, stdev=20.8, Q1=11, median:22, Q3=36.
Summary statistics for number of sentences per file:
D_{14}: average=86.6, stdev=31.1, Q1=69, median:78, Q3=92;
D_{15}: average=72.2, stdev=22.1, Q1=60, median:67, Q3=78.

Training Dataset[4]: We applied the linguistic rules on D_{14}, which labeled 2837 sentences as *adverse_remark* and 9861 as *NA* (rules did not predict any label for the remaining sentences). We randomly selected additional 1000 sentences from D_{14} as *NA*. The baseline classifiers were trained on this automatically created labeled training dataset (denoted D_{TR}). Note that (i) no manual efforts were spent in labeling the sentences in D_{TR}; and (ii) linguistic rules and FinBERT did not use the training dataset D_{TR}.

Evaluation Dataset: From D_{15}, we removed the sentences that did not have a single verb, or had length < 10 or > 50. Out of the remaining sentences we randomly selected 500 sentences making sure they are sufficiently different from each other to avoid near-duplicates. We had designed detailed annotation guidelines using which we manually labelled each sentence as either *adverse_remark* or *NA*. A total of 82 sentences were labeled as *adverse_remark* and the remaining 418 as *NA*. We denote this dataset as D_{EV}. The basic guideline for marking a sentence as *adverse_remark* is already mentioned in Introduction. Examples of other guidelines are as follows. (i) *A sentence containing mention of an action of the management/state of the company which indicates a problem should be marked adverse*. Example: `Company is not in a position to meet its financial obligations.` (ii) *A sentence in which the auditor is only stating facts (without negative opinion) should not be marked as adverse, though the facts may appear negative from a business perspective.*. Using this, the sentence `Company has disclosed the impact of pending litigation in financial statements`, should *not* be marked as *adverse_remark*, because the auditor is noting that the company had taken a correct action. Kappa statistic for inter-annotator agreement (2 annotators) was $\kappa = 0.718$, which is substantial agreement.

[4]Our labeled datasets will be made available upon request.

Method	P	R	F_1
Rules	**0.96**	0.28	0.43
FinBERT	0.37	0.26	0.30
Random Forest (RF)	0.58	0.72	0.65
Logistic Regression (LS)	0.39	0.76	0.51
SVM_RBF	0.46	**0.84**	0.6
XGBoost	0.60	0.50	0.55
M_{BERT}	0.68	0.74	0.71
Ensemble	0.69	0.76	**0.72**

Table 3: Evaluation of Task A.

Evaluation: Baseline classifiers and M_{BERT} were trained on D_{TR}. Then, these classifiers, and other methods mentioned earlier, were applied to predict the labels for the sentences in D_{EV} (Table 3). The precision of the rules is the highest, even when applied to sentences in D_{EV} drawn from D_{15}, which shows that the rules are robust. *SVM* has the highest recall, but relatively poor precision. M_{BERT} outperforms other methods in terms of F_1 measure, and the simple *Ensemble* method has the highest overall performance in terms of F_1. Many errors of M_{BERT} seem to be because of incorrect learning of attention weights. M_{BERT} wrongly identifies `Note 27 describes the uncertainty related to the outcome of the lawsuit filed against the Company.` as *adverse_remark* because it assigns high attention weight to `uncertainty` and this word has appeared more frequently with *adverse_remark* class in the training dataset. M_{BERT} missed the adverse remark `Company has given corporate guarantee in earlier years for loans taken by subsidiary company from a bank.`. This sentence is adverse because of the domain knowledge *giving guarantee for loans by other companies may be risky*. Such adverse remarks are not covered by linguistic rules; so they are absent in D_{TR}.

Performance on Task B

Evaluation Dataset: We manually assigned 0, 1, or more XBRL categories to each of the 82 *adverse_remark* sentence from D_{EV}, since a sentence may contain mentions of many FV. We denote this dataset as D_B. Total of 31 distinct XBRL categories were used in labeling the 82 sentences (average number of XBRL categories per sentence is 1.38). We have prepared detailed guidelines for this annotation. The suggested approach is to identify mentions of FV in the given sentence, and then assign the "nearest" matching XBRL category for each FV mention. A difficulty arises when multiple XBRL categories have a near match for a given FV mention, in which case annotators need to carefully understand the context of that FV mention from the sentence before assigning the XBRL category. Example: `We invite attention to Note no. 29.02 of the financial statements regarding the contingent liability with regard to proceedings under the Income tax Act , 1961 including non - filing of return for the financial year 2013 - 14.` Here FV mentions could be `contingent liability` and `Income Tax`. However, `Income Tax` is part of an act name and

Equity Share Capital	Preference Share Capital	**Total Share Capital**	Reserves and Surplus	**Total Reserves and Surplus**	**Total Shareholders Funds**	Long Term Borrowings	Long Term Provisions	**Total Non-Current Liabilities**	Short Term Borrowings	Trade Payables	Other Current Liabilities	Short Term Provisions	**Total Current Liabilities**
3.12	4.25	**7.37**	-7.36	**-7.36**	**0.01**	0.35	0	**0.35**	0.7	0.06	2.98	0	**3.75**

Total Capital And Liabilities	Tangible Assets	**Fixed Assets**	Deferred Tax Assets [Net]	Long Term Loans And Advances	Other Non Current Assets	**Total Non-Current Assets**	Inventories	Trade Receivables	Cash And Cash Equivalents	Short Term Loans And Advances	**Total Current Assets**	**Total Assets**	Contingent Liabilities
4.11	1.03	**1.03**	2.69	0.1	0	**3.82**	0.04	0.15	0.01	0.08	**0.29**	**4.11**	0.2

Figure 4: An Example Balance Sheet.

Method	All categories			Top 5 frequent		
	P	**R**	**F_1**	**P**	**R**	**F_1**
SBfull	0.06	0.12	0.08	0.00	0.00	0.00
JS	**0.50**	**0.56**	**0.53**	0.95	**0.76**	**0.84**
SBwin	0.30	0.18	0.23	**1.00**	0.16	0.28
BM25	0.26	0.34	0.30	0.91	0.40	0.56
0SHOT-TC	0.09	0.25	0.13	**1.00**	0.28	0.44

Table 4: Evaluation of Task B.

should not be labeled. `contingent liability` refers to a statutory liability pending to be paid and to be provided for, so the label should be provision statutory liabilities.

Evaluation: Since multiple XBRL categories get predicted for each sentence, *micro-average* F_1 is used as the evaluation metric (Table 4). For a method, for each category X, true positives are those sentences for which X is present in evaluation dataset D_B as well as in predicted set. Sentences for which X is not present in D_B but present in predicted set are false positives. Sentences for which X is present in D_B but absent in predicted set are false negatives. Thresholds $\theta_0, \theta_1, \theta_2$ and θ_3 are chosen where highest $F_{1,MicroAvg}$ is seen; m is chosen to get better recall for sentences having many categories. SBfull performs poorly because the encoding obtained by Sentence-BERT is for the entire sentence, with which the embedding of a particular category tends to have low similarity. SBwin performs better because it uses windows over the sentences, rather than the full sentence. 0SHOT-TC (entailment) poorly captures relevance of XBRL category to a given sentence. A common reason for this poor performance is that the language models used for these methods are trained on a different corpus (not financial text) and used here without fine tuning. JS and BM25 perform better for the categories that are likely to be present in audit reports as is; e.g. Fixed Assets, Investments etc. They give false positives where the category name appears but not as a FV, as explained in the earlier example of `Income Tax`. None of the methods is able to correctly predict categories that require domain knowledge; e.g. *Shareholders' funds are part of the net worth of a company*. If the auditor made an adverse remark for net worth, then the category Shareholders' funds is implied based on this knowledge. Despite all the challenges of diverse mentions of categories, unsupervised nature of detection techniques and unavailability of domain knowledge; except for SBfull, all the methods give high precision for top 5 frequent categories in gold standard viz, Payments income taxes, Taxes payable sales tax, Central excise duty, Shareholders' funds and Profit loss account.

Generating Audit Suggestions

Fig. 4 shows an example balance sheet, which was identified as anomalous by an anomaly detection technique (values are in units of Rupees 10 million). The explanation generation technique identified Total Assets, Fixed Assets (among others) as the FV that make this balance sheet anomalous. A set of 16116 sentences out of total 166935 sentences (9.6%) were predicted as *adverse_remark* using the Ensemble method on the dataset $D_{14} \cup D_{15}$. We had removed short sentences, noisy sentences (such as table rows), and duplicate sentences before applying the method. 113 and 127 sentences were labeled with XBRL categories Total Assets, Fixed Assets respectively. From these we can select the top k adverse remarks about each category. Alternatively, the user can specify another text T, which we map to a suitable XBRL category X_T, which can be more general (ancestor) or more specific (descendant) than the above two; e.g., if T = `manufacturing plant buildings` then X_T = Factory Building. Then we can identify adverse remarks which are labeled with X_T, select some of these and transform them (using simple linguistic rules for text transformation) as audit suggestions. For example, the adverse remark sentence `There was disposal of factory building during the year affecting the going concern of the company.` can be selected and transformed into an audit suggestion *"Check if there was disposal of factory building"*.

Conclusions and Future Work

We applied NLP techniques on financial audit reports, to extract the hidden knowledge, and to use it for assisting auditors. We proposed to generate suitable audit suggestions for auditors as additional focus points, apart from their standard operating procedures. The audit suggestions are generated based on knowledge extracted from a large corpus of past audit reports of many companies. This knowledge consists of a set of automatically identified sentences of type adverse remarks, the financial variables mentioned in each sentence and automatically assigned XBRL categories for them. We created a set of high precision linguistic rules to identify adverse remark sentences, created a labeled training data using them, and then trained a classifier to identify such sentences. We also presented several strategies to assign zero, one, or more XBRL categories to any given adverse remark sentence. We evaluated the proposed approaches against competent baselines. Given a company's suspicious financial statements, and a subset of financial variables in them that contribute to suspiciousness, we match these with the extracted knowledge base and identify aligned adverse remarks which can be utilized to generate *audit suggestions*, to help the auditor in focusing on specific directions for further investigations.

For further work, we plan to create a more complex model of audit suggestions which will be more human-understandable, will incorporate auditors' domain-knowledge and will be personalizable for auditors. We are also conducting user studies with real auditors and forensic accounting experts and will incorporate their feedback.

References

Araci, D. 2019. FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Arens, A.; Elder, R.; and Beasley, M. 2016. *Auditing and Assurance Services: An Integrated Approach, 16th ed.* Pearson Publishers.

Ashtiani, M. N., and Raahemi, B. 2021. Intelligent fraud detection in financial statements using machine learning and data mining: a systematic literature review. *IEEE Access* 10:72504–72525.

Basiri, M. E.; Nemati, S.; Abdar, M.; Cambria, E.; and Acharya, U. R. 2021. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems* 115:279–294.

Devine, P., and Blincoe, K. 2022. Unsupervised extreme multi label classification of stack overflow posts. In *2022 IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering (NLBSE)*, 1–8.

Kirkos, E.; Spathis, C.; and Manolopoulos, Y. 2007. Data mining techniques for the detection of fraudulent financial statements. *Expert systems with applications* 32(4):995–1003.

Loukas, L.; Fergadiotis, M.; Chalkidis, I.; Spyropoulou, E.; Malakasiotis, P.; Androutsopoulos, I.; and Paliouras, G. 2022. FiNER: Financial numeric entity recognition for XBRL tagging. *arXiv preprint arXiv:2203.06482*.

Maka, K.; Pazhanirajan, S.; and Mallapur, S. 2020. Selection of most significant variables to detect fraud in financial statements. *Materials Today: Proceedings*.

Nigrini, M. J. 2020. *Forensic Analytics: Methods and Techniques for Forensic Accounting Investigations*. Wiley, 2nd edition.

Patil, S., and Ravindran, B. 2015. Active learning based weak supervision for textual survey response classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 309–320.

Perols, J. 2011. Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory* 30(2):19–50.

Reimers, N., and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Robertson, S., and Zaragoza, H. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval* 3(4):333–389.

Shinde, A.; Vaishampayan, S.; Apte, M.; and Palshikar, G. K. 2022. Unsupervised detection of misinformation in financial statements. In *The International FLAIRS Conference Proceedings*, volume 35.

Vaishampayan, S.; Shinde, A.; Pawde, A.; Pawar, S.; Apte, M.; and Palshikar, G. K. 2022. Explainability for misinformation in financial statements. In *Proceedings of the CIKM AIMLAI 2022 Workshop co-located with 31st ACM International Conference on Information and Knowledge Management (CIKM 2022)*, volume 3318.

Yadav, A.; Jha, C.; Sharan, A.; and Vaish, V. 2019. Sentiment analysis of financial news using unsupervised and supervised approach. In *International Conference on Pattern Recognition and Machine Intelligence*, 311–319. Springer.

Yin, W.; Hay, J.; and Roth, D. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.