

FinTAGGING: Benchmarking LLMs for Extracting and Structuring Financial Information

Yan Wang
The Fin AI
USA

Yang Ren
Lingfei Qian
Xueqing Peng
The Fin AI
USA

Yi Han
Georgia Institute of
Technology
USA

Keyi Wang
Columbia University
USA

Dongji Feng
California State
University
USA

Fengran Mo
University of Montreal
Canada

Shengyuan Lin
Carnegie Mellon
University
USA

Qinchuan Zhang
Rensselaer Polytechnic
Institute
USA

Kaiwen He
Chenri Luo
Jianxing Chen
Junwei Wu
Columbia University
USA

Jimin Huang*
The Fin AI
USA
jimin.huang@thefin.ai

Guojun Xiong
Harvard University
USA

Xiao-Yang Liu
Columbia University
USA

Qianqian Xie
The Fin AI
USA

Jian-Yun Nie
University of Montreal
Canada

Abstract

Accurately understanding numbers from financial reports is fundamental to how markets, regulators, algorithms, and normal people read the economy and the world, yet even with XBRL (eXtensible Business Reporting Language) designed to tag every figure with standardized accounting concepts, mapping thousands of facts to over 10,000 U.S. GAAP concepts remains costly, inconsistent, and error-prone. Existing benchmarks define tagging as flat, single-step, extreme classification over small subsets of US-GAAP concepts, overlooking both the taxonomy's hierarchical semantics and the structured nature of real tagging, where each fact must be represented as a contextualized multi-field output. These simplifications prevent fair evaluation of large language models (LLMs) under realistic reporting conditions. To address these gaps, we introduce FinTAGGING, the first comprehensive benchmark for structure-aware and full-scope XBRL tagging, designed to evaluate LLMs' ability to extract and align financial facts through numerical reasoning and taxonomy alignment across text and tables. We define two subtasks: FinNI for numeric identification, which extracts numerical entities and their types from XBRL reports, and FinCL for concept linking, which maps each extracted entity to the corresponding concept in the full US-GAAP taxonomy. Together, these subtasks produce a structured representation of each financial fact. We evaluate diverse LLMs under zero-shot settings and analyze

their performance across both subtasks and overall tagging accuracy. Results show that LLMs generalize well in numeric identification but struggle with fine-grained concept linking, revealing current limitations in structure-aware reasoning for accurate financial disclosure. All code and datasets are available on GitHub¹ and Hugging Face².

CCS Concepts

- Information systems → Test collections; Information extraction; Clustering and classification.

Keywords

XBRL tagging, Benchmark, Large language model, Information extraction, Information retrieval, Reranking

ACM Reference Format:

Yan Wang, Yang Ren, Lingfei Qian, Xueqing Peng, Yi Han, Keyi Wang, Dongji Feng, Fengran Mo, Shengyuan Lin, Qinchuan Zhang, Kaiwen He, Chenri Luo, Jianxing Chen, Junwei Wu, Jimin Huang, Guojun Xiong, Xiao-Yang Liu, Qianqian Xie, and Jian-Yun Nie. 2018. FinTAGGING: Benchmarking LLMs for Extracting and Structuring Financial Information. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 19 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Automated tagging is essential in financial reporting, converting structured content such as tables and text into machine-readable data by linking each number to its meaning. Globally, over 2 million companies publish financial reports disclosing earnings, expenses, and liabilities, which provide essential information for investors, regulators, and analysts. However, inconsistent terminology poses challenges for reliable interpretation. To address this, the eXtensible Business Reporting Language (XBRL) was introduced in 1999

¹<https://github.com/The-FinAI/FinTagging/>

²<https://huggingface.co/collections/TheFinAI/fintagging-68270132372c6608ac069bef>

*Corresponding Author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/XXXXXXX.XXXXXXX>

as a global standard for machine-readable financial data [21]. By applying structured tags, as shown in Figure 1, XBRL transforms fragmented disclosures into a consistent format, enabling seamless data exchange, cross-entity comparisons, and large-scale analysis. However, accurately tagging over 2,000 numerical facts per report to more than 10,000 standardized concepts remains labor-intensive and error-prone. In 2023 alone, over 6,500 tagging errors were identified across 33,000 SEC filings³.

Existing approaches to automated XBRL tagging rely on task-specific models and pretrained language models (PLMs) [14, 24], but they require costly annotations, struggle to generalize to new filings, and perform poorly as taxonomies scale. In contrast, recent large language models (LLMs) show strong zero- and few-shot reasoning without task-specific supervision [3, 19, 31, 33], yet their application to XBRL tagging remains underexplored due to two major limitations (Table 1). First, **task framing is oversimplified**: prior benchmarks like FiNER [14] and FNXL [24] treat tagging as extreme multi-class classification over flat label sets, offering little context for disambiguating fine-grained financial facts. These datasets typically cover only 1k+ concepts, leaving most of the 10k+ US-GAAP taxonomy untested. As shown in Table 8, SOTA LLMs such as DeepSeek-V3 [13] and GPT-4o [11] achieve 0.0 precision, recall, and F1 under this setting. Second, **structured data is ignored**: existing datasets exclude tables, despite their central role in financial reporting. Many key facts appear in structured formats, as shown in recent financial QA benchmarks [4, 16, 36], making this omission a key gap between benchmarks and real-world tagging needs.

To address these issues, we present **FINTAGGING**, the first LLM-ready benchmark for full-scope, structure-aware XBRL tagging, which challenges models to perform end-to-end fact extraction and taxonomy alignment on corporate filings. As shown in Table 1, unlike prior work [14, 23, 24] that frames tagging as flat multi-class classification and focuses solely on narrative text, **FINTAGGING** requires models to jointly extract financial facts and align them with the US-GAAP taxonomy across both unstructured text and structured tables. This is the first LLM-oriented formulation that scales to the full set of 10k+ taxonomy labels while avoiding the intractability of single-step classification, enabling more fine-grained evaluation of model performance. In collaboration with financial reporting experts, we decompose the task into two components: the first, **structured numerical information extraction**, assesses a model’s ability to identify key financial facts from both text and tables; the second, **fine-grained concept linking**, evaluates its ability to map each fact to the correct taxonomy concept among all semantically similar candidates. Unlike prior benchmarks limited to 1,000 frequent concepts, our setting covers the entire taxonomy, exposing the limitations of frequency-biased classification and demanding precise semantic reasoning [2]. We construct two evaluation sets: *FinNI-eval*⁴ for numerical fact extraction and *FinCL-eval*⁵ for concept linking, both derived from real XBRL submissions and annotated with gold-standard mappings. We also introduce a unified evaluation framework that jointly measures extraction accuracy and concept alignment, providing a rigorous zero-shot

assessment of LLMs’ capabilities for practical, table-aware XBRL tagging.

We evaluate 13 state-of-the-art LLMs under a zero-shot setting on three fronts: (1) end-to-end macro-F1 over the full **FINTAGGING** benchmark, (2) subtask-specific performance on FinNI and FinCL, and (3) ablation of our unified extraction-and-alignment evaluation framework. DeepSeek-V3 [13] and GPT-4o [11] achieve the highest macro-F1 scores, indicating that our benchmark design enables strong handling of both frequent and rare financial tags, an improvement over traditional PLM baselines evaluated under token-classification settings. In subtask analyses, these LLMs excel at numerical information extraction (FinNI) but continue to struggle with precise concept linking (FinCL), underscoring the remaining challenge of fine-grained semantic alignment. Crucially, our ablation study shows that, without the joint evaluation framework, even top LLMs produce invalid tagging outputs, highlighting the framework’s essential role in realistic assessment of XBRL tagging readiness.

We conclude our main contributions as follows: (1) We introduce **FINTAGGING**, the first LLM-ready benchmark for full-scope, table-aware XBRL tagging with a two-stage pipeline, so that modern LLMs can be probed in a true zero-shot setting. For each stage, we release a new, professionally annotated evaluation set: FinNI-eval and FinCL-eval. (2) We conduct an extensive evaluation of state-of-the-art LLMs in zero-shot settings, systematically analyzing their performance across information extraction and semantic alignment subtasks, as well as overall tagging accuracy. (3) Our experimental results reveal a substantial performance gap between LLMs and the task requirements, particularly in fine-grained semantic alignment. This highlights the limitations of current LLMs in complex financial applications and underscores the need for continued advancements in task-specific adaptation.

2 Related Work

XBRL Tagging Benchmarks: Previous benchmarks have framed XBRL tagging as a large-scale classification problem, primarily focusing on entity recognition. FiNER [14] introduced a dataset of 1.1 million sentences from SEC filings, annotated with 139 XBRL entity types, addressing challenges like context-sensitive numeric entities and financial domain-specific expressions. Building on this, Sharma et al. [24] proposed the Financial Numeric Extreme Labelling (FNXL) dataset, expanding the label space to approximately 2,800 XBRL tags and employing extreme classification techniques to handle the increased scale. While these works emphasize the complexity of financial taxonomies, their flat extreme classification approach overlooks the reasoning and alignment needed for realistic XBRL tagging, limiting their suitability for evaluating large language models.

XBRL Tagging Methods: To improve XBRL tagging accuracy, prior work has incorporated structured knowledge and domain-specific modeling. Saini et al. [22] proposed GalaXC, a graph neural network leveraging label hierarchies for improved classification. Ma et al. [15] enriched transformer models with taxonomy definitions to disambiguate semantically similar tags. Loukas et al. [14] introduced SECBERT and numeral-aware input transformations to enhance robustness. For extreme multi-label tagging, Sharma

³<https://xbrl.us/wp-content/uploads/2023/03/DQC-SECMeetingNotes-20240314.pdf>

⁴<https://huggingface.co/datasets/TheFinAI/FinNI-eval-v1>

⁵<https://huggingface.co/datasets/TheFinAI/FinCL-eval-v1>

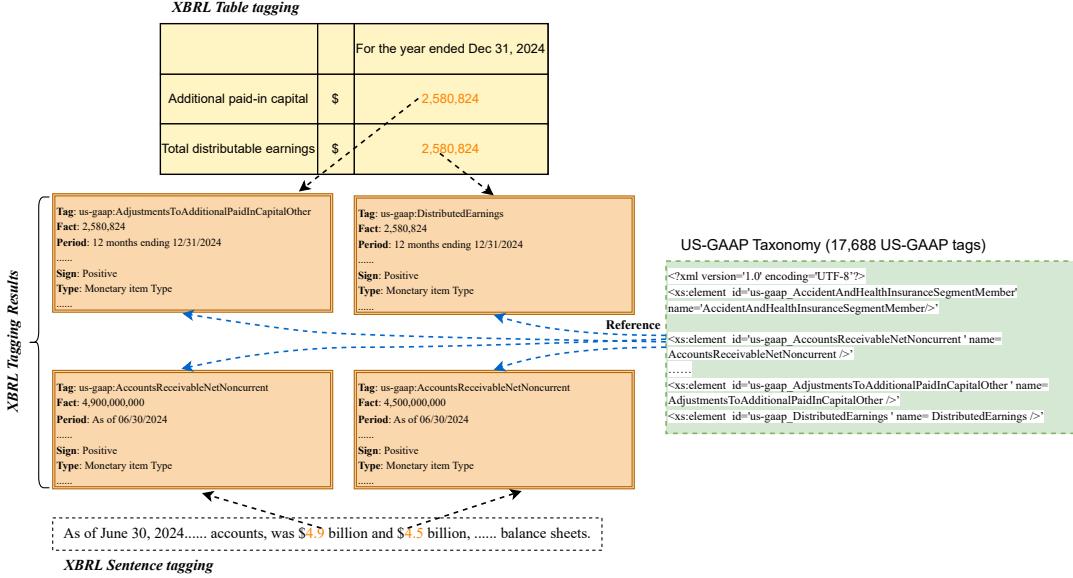


Figure 1: An example of Financial Tagging in Realistic, where the orange numbers mark facts to be tagged, black dashed arrows show the structured outputs after tagging, and blue dashed arrows denote the US-GAAP taxonomy referenced during tagging.

Table 1: Detailed comparison of financial NLP benchmarks across task types, sources, and structural capabilities. “Num.Reasoning” indicates the numerical reasoning. “Struct.IE” denotes the structured information extraction.

Benchmark	Scenario	Data Source	Task	Modality	#Entity Label	#Taxonomy Label	Num. Reasoning?	Struct. IE?	Concept Linking?
FinQA [4]	Decision making	SEC 10-K	QA	text/table	0	0	✓	✗	✗
ConvFinQA [5]	Decision making	SEC 10-K	QA	text/table	0	0	✓	✗	✗
TAT-QA [36]	Financial analysis	Chinese financial reports	QA	text/table	0	0	✓	✗	✗
DocVQA [16]	Enterprise automation	Financial document images	QA	text/image	0	0	✓	✗	✗
FiNER-ORD [23]	Financial tagging	Chinese financial disclosures	Classification	text	49	0	✗	✗	✗
FinRED [25]	Knowledge graph construction	English financial news	Classification	text	38	0	✗	✗	✗
FiNER [14]	XBRL tagging	SEC 10-K	Extreme classification	text	0	139 / 17,388	✓	✗	✗
FNXL [24]	XBRL tagging	SEC 10-K	Extreme classification	text	0	2,800 / 17,388	✓	✗	✗
FinTagging (ours)	Financial & XBRL tagging	SEC 10-K	IE + Alignment	text/table	5	17,388 / 17,388	✓	✓	✓

et al. [24] applied AttentionXML to focus on context-relevant segments. Wang [30] aligned custom tags with standard taxonomies via semantic similarity using TF-IDF, Word2Vec, and FinBERT. More recently, Han et al. [10] developed XBRL-Agent, an LLM-based system for extracting insights from filings. These efforts highlight the shift toward taxonomy-aware and LLM-enhanced solutions for XBRL tagging.

Financial Evaluation Benchmarks: Broader financial NLP benchmarks address information extraction, reasoning, and document understanding tasks. FiNER-ORD [23] and FinRED [25] focus on entity recognition and relation extraction, respectively. BizBench [12] assesses the quantitative-reasoning ability of LLMs for both business and finance. Pixiu [35] evaluates LLMs across classification, QA, and summarization, while FinQA [4] and ConvFinQA [5] focus on numerical reasoning over text and tables. TAT-QA [36] and DocVQA [16] target table-text joint understanding. These benchmarks highlight the importance of integrating structured and unstructured information, yet they overlook the taxonomy-driven fact alignment and do not support the structured output required for XBRL tagging.

3 FINTAGGING

3.1 Task Formulation

Formally, given a financial document D consisting of multiple textual contents and structured tables, a set of predefined value data types L specified by the XBRL specification, and a taxonomy database \mathcal{T} containing a set of financial semantic concepts, the XBRL tagging task is to identify all relevant numerical values in D and annotate each with a structured triplet $\{\text{Fact}, \text{Type}, \text{Tag}\}$. Here, Tag denotes the linked concept in \mathcal{T} (e.g., “us-gaap:CashAndDueFromBanks”, etc.), Fact represents the extracted numerical value as it appears in context (e.g., “2.5”, “10,000”, “two”, etc.), and Type specifies the corresponding data type in L (including *monetaryItemType*, *percentItemType*, *sharesItemType*, *perShareItem*, and *integerItemType*). Each numerical fact is assigned a single, most specific tag from the taxonomy, without overlapping. Specifically, *monetaryItemType* refers to a financial amount, such as revenue or expenses, typically reported in currency units (e.g., USD). *percentItemType* represents a rate or ratio, expressed as a decimal (e.g., 25%). *sharesItemType*

indicates the number of shares, such as stocks held or issued. *perShareItemType* captures values reported on a per-share basis, like earnings per share. Lastly, *integerItemType* is used for whole number counts, such as the number of employees or transactions.

We formalize the task as a mapping:

$$f : (D, L, \mathcal{T}) \mapsto \{(\text{Fact}_i, \text{Type}_i, \text{Tag}_i)\}_{i=1}^n \quad (1)$$

where each triplet corresponds to a financial value mention extracted from D and semantically grounded in \mathcal{T} .

Inspired by information extraction and alignment works [18, 29, 32, 34] and after discussed with financial reporting specialists, we formulated **FINTAGGING** into two sub-tasks: **Financial Numeric Identification (FinNI)**, a multi-modal numerical information extraction task, detects numerical value in D and classifies each with its appropriate Type. **Financial Concept Linking (FinCL)**, a numerical entity normalization task, then associates each identified value with its most appropriate Tag in \mathcal{T} based on contextual and structural cues.

Financial Numeric Identification (FinNI). The first subtask of **FINTAGGING** focuses on identifying numerical values in a financial document and assigning each a coarse-grained value data type. This corresponds to detecting the Fact and Type components of each triplet $\{\text{Fact}, \text{Type}, \text{Tag}\}$ defined in the overall task.

We formalize this subtask as a mapping:

$$f_{\text{FinNI}} : (D = (S, T), L) \mapsto \{(e_i, l_i)\}_{i=1}^k \quad (2)$$

where S and T represent the textual and tabular components of the document, respectively, and L is the set of predefined value data types. Each e_i is a numerical entity extracted from either S or T , and $l_i \in L$ denotes its assigned data type.

Financial Concept Linking (FinCL). Building on the output of the FinNI subtask, the goal of FinCL is to semantically ground each identified numerical entity e by linking it to a concept \hat{c} in a predefined financial taxonomy, which is the Tag component of each triplet $\{\text{Fact}, \text{Type}, \text{Tag}\}$ defined in the overall task. Formally, we define the mapping as:

$$f_{\text{FinCL}} : (e, l, C_e, \mathcal{T}) \mapsto \hat{c} \quad (3)$$

where e is a numerical entity identified in the document $D = (S, T)$, $l \in L$ is its predicted data type from FinNI, C_e denotes the contextual information surrounding e in D , and $\mathcal{T} = \{c_1, c_2, \dots, c_n\}$ is a financial taxonomy containing n uniquely defined and semantically grounded concepts. The model is required to assign a concept $\hat{c} \in \mathcal{T}$ that best reflects the meaning of e in its structural context.

3.2 Raw Data Collection

We compiled 142 annual 10-K reports filed in 2023 and 2024 by publicly listed companies from the SEC⁶, as shown in Table 2. To ensure diversity, the collection covers all 11 major industry sectors, with about 95% of companies distributed across 31 U.S. states and the remaining 5% based outside the United States. Using BeautifulSoup, we parsed these reports and extracted 319,893 narrative sentences (roughly 76 million characters) along with 21,576 financial tables. Company-level details are provided in Appendix D.2.

⁶<https://www.sec.gov/>

Each report follows the SEC's XBRL filing standard, meaning that the underlying *instance*, *schema*, and *linkbase* documents already contain machine-readable tags (ground-truth tags) that link textual and tabular values to US-GAAP taxonomy concepts. We included all sections that may reference or embed XBRL-tagged content to ensure comprehensive coverage. For the justification of our benchmark granularity, please see the Appendix E.

Table 2: Statistics of collected financial reports.

Item	Value
Report type	10-K
Period	2023-02-13 to 2025-02-13
#Company	142
#Covered sector	11
#Covered jurisdiction	31 States + outside US
#Sentence	319,893
#Char	75,748,949
#Table	21,576

3.3 Data Curation

To construct the **FINTAGGING** benchmark and ensure both efficiency and fairness in evaluation, we designed a **two-stage filtering and annotation pipeline**. This pipeline systematically processes the raw 10-K filings to retain representative, high-quality instances that collectively cover all 142 companies, five valid entity types, and every numerical US-GAAP concept observed in the corpus. The overall workflow is illustrated in Algorithms 1 and 2.

Stage 1: Filtering Raw Instances (Algorithm 1). The first stage aims to remove noisy or redundant samples while ensuring comprehensive coverage of both entity types and US-GAAP concepts. Each narrative sentence or table instance is automatically examined by a rule-based parser that checks if it contains numeric entities tagged with valid US-GAAP concepts. The parser performs schema-level consistency checks based on the taxonomy definitions to ensure tagging correctness. Instances without valid entities are classified as NEG (negative). Those containing valid entities are treated as positive and further divided into two categories: (1) POS_SELECT, which introduces at least one new US-GAAP concept not yet covered; and (2) POS_CANDIDATE, which contains only previously covered concepts. This design ensures that each concept appears at least once in the dataset while minimizing redundancy across companies and reports.

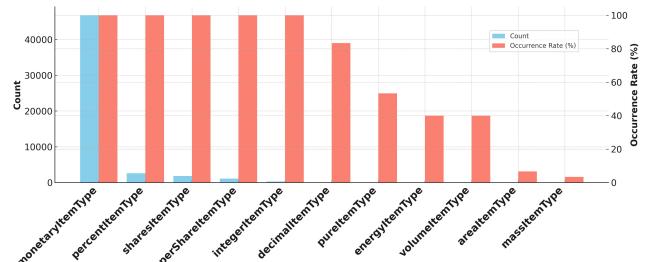


Figure 2: Statistics of numerical entity types.

To determine which entity types should be considered valid, we first profiled all XBRL numeric item types that appeared in the raw

Algorithm 1: Screening of Text and Table Instances

Input: 10-K reports; US-GAAP taxonomy; 5 valid entity types
Output: POS_SELECT, POS_CANDIDATE, NEG

Initialize: an empty set of covered concepts; empty lists for POS_SELECT, POS_CANDIDATE, and NEG;

foreach *instance (texts or tables) in 10-K reports do*

- Check if the instance contains numeric entities that are both: (i) tagged with a US-GAAP concept, and (ii) of a valid entity type;
- if no such entities are found then**
 - Add the instance to NEG;
- else**
 - Identify all US-GAAP concepts in this instance;
 - if the instance introduces at least one new concept not yet covered then**
 - Add the instance to POS_SELECT and update the covered concepts set;
 - else**
 - Add the instance to POS_CANDIDATE;

return {POS_SELECT, POS_CANDIDATE, NEG};

corpus and summarized the results in Figure 2. For each type, we calculated both its cross-company coverage (the number of companies in which it occurs) and its overall frequency (the proportion of all annotated numeric entities belonging to that type). The figure shows a clear long-tailed distribution: a total of eleven numeric item types were identified, but only a few dominate in both coverage and frequency. The five types that consistently appear across nearly all companies and collectively account for the majority of entities are *monetaryItemType*, *percentItemType*, *sharesItemType*, *perShareItemType*, and *integerItemType*. These five types form the label space for all subsequent stages. Under this definition, Algorithm 1 treats an instance as a potential positive only if it contains at least one US-GAAP-tagged entity belonging to this label space; otherwise it is assigned to NEG.

To maintain structural fidelity, additional validity rules are applied. For narrative texts, we retain only segments longer than 20 characters that contain numerical tokens. For tables, we preserve the *<table>*, *<tr>*, *<td>*, and *<th>* tags to ensure layout integrity, while tables with headers only (i.e., without numeric values) are excluded.

Stage 2: Company-Capped Sampling and Minimal Cover (Algorithm 2). The second stage refines the positive pool by enforcing both per-company balance and concept coverage completeness. For each modality (text or table), we first perform company-capped seeding: from each company’s POS_SELECT, up to $K = 10$ positive instances are retained as the seed set, while the remaining positives are moved into a candidate pool. We then mark all US-GAAP concepts covered by the seed and identify those still uncovered. Finally, a greedy minimal-cover procedure is executed: candidates are iteratively selected according to the number of uncovered concepts they can cover, with ties broken by shorter text length, fewer entities, or earlier order. Each selected candidate is added to the final set P^* , and the process continues until all concepts are covered or the candidate pool is exhausted. The resulting positive set P^* thus maintains full US-GAAP concept coverage, balanced company

Algorithm 2: Company-Capped Seeding + Minimal Cover (for Text and Table)

Input: POS_SELECT grouped by company (run separately for text and table) from algorithm 1;
All concepts observed in the POS_SELECT;
Cap $K = 10$ (maximum positives per company for seeding)
Output: Final positive instances P^* and leftover instances

Seed selection. For each company, keep up to K positives as the seed set. Put the rest into a new candidate pool.

Check coverage. Mark all concepts already covered by the seed. Identify which concepts are still uncovered.

Greedy covering. While uncovered concepts remain:

- (1) For each candidate, count how many uncovered concepts it can cover.
- (2) Pick the candidate with the highest count (ties broken by shorter text, fewer entities, or earlier order).
- (3) Add the candidate to the final positives, update the covered concepts, and remove it from the pool.

Finalize. The final positives P^* are the union of the seed and the greedy additions. The leftover candidates form the candidate set.

representation, and reduced redundancy, forming a compact yet coverage-complete dataset.

Final Dataset and Validation. After two-stage filtering, we obtained 3,084 textual sentences and 3,067 table sequences from financial filings (Table 3). Among them, 2,054 text and 2,045 table instances are positives containing valid XBRL tagging information, while 1,030 text and 1,022 table instances are negatives. Textual inputs contain on average 96.25 tokens, whereas table sequences are much longer (1,204.72 tokens on average). Overall, the dataset includes 62,668 annotated entities linked to 3,953 unique US-GAAP concepts across the five valid entity types.

Based on this foundation, we further derive two subtask-specific datasets: **FinNI-eval**, targeting numerical entity identification, and **FinCL-eval**, targeting concept linking. Both are the first datasets in the financial domain to emphasize structured information extraction and concept-level alignment. Their detailed statistics are reported in Table 4.

To verify annotation reliability, two domain experts independently reviewed 100 randomly sampled entries. The results indicate a 96% raw agreement and a Cohen’s Kappa of 81.13, confirming the robustness and reproducibility of our semi-automated annotation pipeline. The validation guideline is provided in Appendix F, and a complete unfiltered version of the benchmark is included in Appendix D.1.

3.4 FinNI-eval Dataset Construction

Based on the annotation procedure described in Section 3.3, we construct the FinNI-eval dataset, comprising 6,151 instances derived from the annotated sentences and tables. Specifically, the dataset comprises two components: the input block and the answer block. The input block includes a task instruction and the input content. The instruction defines the FinNI task by specifying the responsibilities of the LLMs, the definitions of entity types, and the rules to be followed. The answer block is formatted as a list aligned with the input block. The answer is a JSON list of all identified entity values

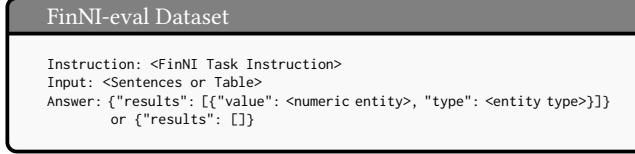
Table 3: Statistical information for the original dataset in our benchmark. Tokens are calculated using the “cl100k_base” tokenizer (\pm standard deviation).

Item	Sentence	Table
Positive instances	2,054	2,045
Negative instances	1,030	1,022
Avg. Tokens/S	96.25 ± 72.75	$1,204.72 \pm 1,162.41$
Avg. Entities/S	1.88 ± 2.11	18.54 ± 29.14
Avg. Concepts/S	1.88 ± 2.11	18.54 ± 29.14
Total Entities		62,668
Entity Types		5
Unique Concepts		3,953

Table 4: Statistics of evaluation datasets for FinNI-eval and FinCL-eval. Tokens are calculated using the “cl100k_base” tokenizer (\pm standard deviation).

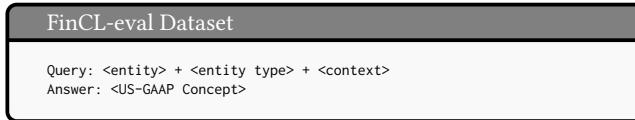
Item	FinNI-eval	FinCL-eval
#Instance	6,151	62,668
Avg. Input Tokens	1101.50 ± 992.00	60.43 ± 48.25
Max Input Tokens	18,831	750
Avg. Output Tokens	175.24 ± 375.90	12.82 ± 5.09
Max Output Tokens	7,813	41

and types or an empty JSON list. The specific prompt template is shown in Figure 9 in the Appendix.



3.5 FinCL-eval Dataset Construction

Following the formulation of the FinCL subtask in Section 3.1, we further construct a FinCL-eval dataset that includes 62,668 query-answer pairs for numerical entity normalization. In addition, we built a US-GAAP taxonomy database containing 17,688 unique financial concepts. To create the FinCL-eval dataset, we utilize all positive instances identified in Section 3.3. Each query consists of a numerical entity, its corresponding entity type, and its surrounding context, while the answer is the associated US-GAAP concept.



3.6 Evaluation

3.6.1 Evaluation Framework. We propose a unified evaluation framework for the **FINTAGGING** benchmark to assess LLM performance. As shown in Figure 3, given a financial report D , the framework reformulates tagging into two subtasks, FinNI and FinCL, to generate structured triplets {Fact, Type, Tag}. This design jointly evaluates an LLM’s zero-shot ability in fact extraction and concept alignment. For fair comparison with token-classification baselines, we report macro and micro Precision, Recall, and F1 [26]. FinNI is evaluated using pair-level metrics, while FinCL uses accuracy. Full metric definitions are provided in Appendix G.

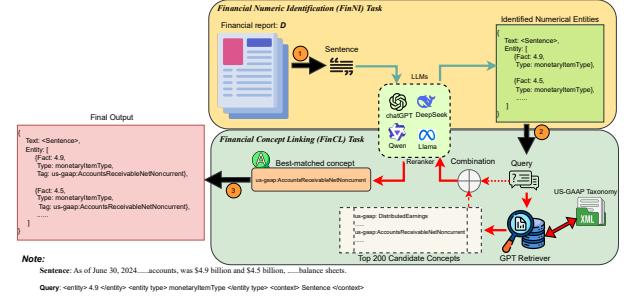


Figure 3: A unified evaluation framework on **FINTAGGING benchmark. Note: The Sentence is drawn from financial report D , and the Query provides an example that incorporates the entity 4.9. The black arrows show the overall framework flow, green arrows denote the FinNI process, red arrows indicate the FinCL process, red bidirectional arrows represent retrieval interactions with the taxonomy, and red dashed arrows mark the recomposed input between the query and retrieved candidates.**

FinNI Evaluation: We use **pair-level metrics** to evaluate the extraction performance on FinNI. The objective is to evaluate the LLM’s ability to distinguish between structured and unstructured contexts and to extract financial values based on its semantic understanding and domain-specific knowledge.

FinCL Evaluation: To assess whether LLMs can perform fine-grained XBRL entity normalization, **we reformulate the task as a retrieval-reranking problem**, where LLMs are used to disambiguate and select the most appropriate taxonomy concept (reranking stage) from a reduced candidate set (retrieval stage), avoiding the impracticality of direct multi-thousand-way classification (prompt template used for reranking in the FinCL subtask is shown in Figure 10 in Appendix).

To this end, we first generate embeddings for each taxonomy concept using `text-embedding-3-small`⁷, and retrieve the top- h candidate tags from the US-GAAP taxonomy \mathcal{T} based on semantic similarity. Therefore, the LLMs are required to rerank the retrieved candidates by leveraging deeper contextual understanding to select the best-matched tag $\hat{c} \in \mathcal{T}_h$, formalized as:

$$f_{\text{rerank}} : (e, l, C_e, \mathcal{T}_h) \mapsto \hat{c} \quad (4)$$

3.6.2 Evaluation Models. Our goal is to evaluate the foundational capabilities of SOTA LLMs on the **FINTAGGING** benchmark, to better understand their strengths and limitations in financial tagging. To this end, we evaluate models across three categories using our proposed evaluation framework, as detailed in Table 13. These include: (1) one general-purpose closed-source LLM: GPT-4o model [11]; (2) eleven general-purpose open-source LLMs including DeepSeek-V3 [13], DeepSeek-R1-Distill-Qwen-32B [6], Qwen3-32B [28], Qwen3-14B [28], Qwen3-1.7B [28], Qwen3-0.6B [28], Llama-4-Scout-17B-16E-Instruct [17], Llama-3.3-70B-Instruct [9],

⁷<https://platform.openai.com/docs/models/text-embedding-3-small>

Llama-3.1-8B-Instruct [9], Llama-3.2-3B-Instruct [9], and gemma-2-27b-it [27] models; and (3) one domain-specific financial LLM: Fino1-8B [20] model. In addition, we compare these LLMs with strong PLMs, including BERT-large [7], FinBERT [1], and SECBERT [14].

3.6.3 Evaluation setting. We use the LM Evaluation Harness [8] to build customized benchmark suites. Proprietary models are accessed via APIs, while open-source models are evaluated locally on a $4 \times$ H100 GPU cluster (80GB each). We standardize the input length to 2,048 tokens for FinNI and 4,096 for FinCL, with a generation limit of 1,024 tokens to support reasoning-intensive outputs. For fine-tuning strong PTMs such as BERT-large, FinBERT, and SECBERT, we curate 10 additional annual reports as training data (see Appendix I). All data follow the FiNER [14] and FNXL [24] formats. For retrieval in FinCL, we use ElasticSearch⁸ with text-embedding-3-small as the embedding model. For evaluation, we spent a total of 500 GPU hours and about \$1,500 for GPT (from OpenAI) and DeepSeek (from TogetherAI) API.

4 Experiment and Result

4.1 Overall Results

Table 5 presents the overall performance on the FINTAGGING benchmark. It clearly demonstrates that under our framework, LLMs can effectively handle both frequent and rare financial tags, indicating their ability to mitigate long-tail label challenges and underscoring the advantage of our information extraction and alignment formulation over traditional token-level classification approaches.

Table 5: Overall Performance. Bolded values indicate the best performance, underlined values represent the second-best, and italicized values denote the third-best performance.

Category	Models	Macro			Micro		
		P	R	F1	P	R	F1
Closed-source LLM	GPT-4o	0.0865	0.0626	0.0569	0.1052	0.0731	0.0863
	DeepSeek-V3	0.0949	0.0778	0.0659	0.1074	0.1264	0.1162
	Llama-4-Scout-17B-16E-Instruct	0.0683	0.0414	0.0400	0.1045	0.0526	0.0699
	Llama-3.3-70B-Instruct	0.0544	0.0279	0.0288	0.0665	0.0382	0.0485
	DeepSeek-R1-Distill-Qwen-32B	0.0532	0.0283	0.0285	0.0814	0.0214	0.0339
	Qwen3-32B	0.0639	0.0314	0.0324	0.1127	0.0230	0.0382
Open-source LLMs	gemma-2-27b-it	0.0471	0.0291	0.0276	0.0533	0.0390	0.0451
	Qwen3-14B	0.0591	0.0274	0.0288	0.1069	0.0182	0.0311
	Llama-3.1-8B-Instruct	0.0345	0.0178	0.0169	0.0575	0.0166	0.0258
	Llama-3.2-3B-Instruct	0.0194	0.0118	0.0100	0.0178	0.0084	0.0114
	Qwen3-1.7B	0.0207	0.0067	0.0080	0.1063	0.0031	0.0060
	Qwen3-0.6B	0.0026	0.0008	0.0010	0.0562	0.0002	0.0005
Financial LLM	Fino1-8B	0.0344	0.0143	0.0151	0.0419	0.0128	0.0197
Fine-tuned PLMs	BERT-large	0.0252	0.0266	0.0205	0.1518	<u>0.1283</u>	<u>0.1391</u>
	FinBERT	0.0046	0.0064	0.0042	0.0872	0.0526	0.0656
	SECBERT	0.0231	0.0295	0.0203	0.1870	0.1697	<u>0.1779</u>

From the macro-level perspective, which emphasizes balanced performance across both frequent and rare tags, DeepSeek-V3, GPT-4o, and Llama-4-Scout-17B-16E-Instruct achieve the top 3 macro-F1 scores (0.0659, 0.0569, and 0.0400), outperforming all fine-tuned PLMs. This highlights the strong generalization of large LLMs and the effectiveness of our task design. Qwen3-32B also achieves a solid macro-F1 (0.0324), suggesting that good architecture and pre-training can help smaller models perform well in zero-shot settings. From the micro-level perspective, which favors frequent labels, DeepSeek-V3 again performs strongly with a micro-F1 of 0.1162, ranking third overall despite no fine-tuning. GPT-4o also performs

⁸<https://www.elastic.co/>

competitively with a score of 0.0863, outperforming most open-source and domain-specific models.

4.2 Subtask Results

4.2.1 FinNI subtask. Table 6 presents the results of different models on the FinNI subtask. Overall, larger models demonstrate stronger capabilities in identifying numerical entities and producing structured outputs, even without financial domain adaptation. Among them, DeepSeek-V3 achieves the best performance with both the highest recall (0.8430) and F1 (0.6932), while also maintaining competitive precision. In comparison, GPT-4o, the closed-source baseline, performs relatively well with balanced scores, securing the second-best F1 (0.5397). Interestingly, Qwen3-32B and Qwen3-14B yield very high precision (0.6991 and 0.6912, respectively), but their recall is extremely low, leading to weak F1 scores, which highlights that precision alone is insufficient in this task.

Table 6: Performance comparison of different models on the FinNI subtask. Bolded values indicate the best performance, underlined values represent the second-best, and italicized values denote the third-best performance.

Category	Model	Precision	Recall	F1
Closed-source LLM	GPT-4o	0.5893	0.4977	0.5397
	Deepseek-V3	0.5886	0.8430	0.6932
	Llama-4-Scout-17B-16E-Instruct	0.4668	0.3164	0.3771
	Llama-3.3-70B-Instruct	0.4826	0.3301	0.3920
	DeepSeek-R1-Distill-Qwen-32B	0.5676	0.1942	0.2894
	Qwen3-32B	<u>0.6991</u>	0.1804	0.2868
Open-source LLMs	gemma-2-27b-it	0.5060	0.4526	0.4778
	Qwen3-14B	<u>0.6912</u>	0.1487	0.2448
	Llama-3.1-8B-Instruct	0.3874	0.1761	0.2421
	Llama-3.2-3B-Instruct	0.1856	0.1203	0.1460
	Qwen3-1.7B	0.7362	0.0281	0.0541
	Qwen3-0.6B	0.2803	0.0019	0.0038
Financial LLM	Fino1-8B	0.3431	0.1293	0.1878

Smaller open-source models such as Qwen3-1.7B and Qwen3-0.6B show severe limitations, with F1 below 0.1, underscoring the difficulty of the FinNI subtask in zero-shot settings. In contrast, models like gemma-2-27b-it achieve relatively balanced results, ranking third in both recall and F1. Finally, the domain-specific Fino1-8B, despite being fine-tuned on financial reasoning QA, underperforms compared to general-purpose LLMs. This suggests that domain pretraining on financial text alone provides limited benefit unless the training task aligns closely with FinNI's requirements.

4.2.2 FinCL subtask. Table 7 reports the accuracy of different models on the FinCL subtask, which is performed on candidate sets obtained from the best top-200 retrieval results, as shown in Table 16 (Appendix J). Although the retrieval stage provides a foundation, the low Acc@k values, especially for table-based entities, directly constrain the FinCL performance.

Within this reranking formulation, DeepSeek-V3 achieves the best accuracy at 0.1889, closely followed by GPT-4o at 0.1829 and Llama-4-Scout-17B-16E-Instruct at 0.1649. Other large open-source models perform notably worse, with Llama-3.3-70B at 0.1318. The medium-scale models, such as Qwen3-32B (0.1277), DeepSeek-R1-Distill-Qwen-32B (0.1141), gemma-2-27B-it (0.1099), and Qwen3-14B (0.1144), show similar levels. While smaller models, including

Table 7: Performance comparison of different models on the FinCL subtask. Bolded values indicate the best performance, underlined values represent the second-best, and italicized values denote the third-best performance.

Category	Model	Accuracy
Closed-source LLM	GPT-4o	<u>0.1829</u>
Open-source LLMs	Deepseek-V3	0.1889
	Llama-4-Scout-17B-16E-Instruct	<u>0.1649</u>
	Llama-3.3-70B-Instruct	0.1318
	DeepSeek-R1-Distill-Qwen-32B	0.1141
	Qwen3-32B	0.1277
	gemma-2-27b-it	0.1099
	Qwen3-14B	0.1144
	Llama-3.1-8B-Instruct	0.0913
	Llama-3.2-3B-Instruct	0.0415
	Qwen3-1.7B	0.0735
Financial LLM	Fino1-8B	0.0807

Llama-3.2-3B and Qwen3-0.6B, perform near random with 0.0415 and 0.0414. The domain-specific Fino1-8B also lags at 0.0807, suggesting that domain pretraining without explicit task alignment offers little benefit. As a result, overall accuracy remains low, reflecting both the inherent difficulty of handling complex taxonomies and subtle financial semantics, and the cascading effect of retrieval errors that further lowers FinCL performance.

4.3 Ablation analysis

We further compare our benchmark against extreme multi-class classification settings. The prompt template is shown in Figure 11 in the Appendix, and we select the best-performing model from each category (closed-source, open-source, and financial LLMs). The evaluation uses triplet-level (<{Tag, Fact, Type}>) Precision, Recall, and F1.

Table 8: Performance comparison between with/without our evaluation framework on the `FINTAGGING` benchmark dataset.

Evaluation Mode	Model	Precision	Recall	F1
FinTagging	GPT-4o	0.1166	0.0915	0.1026
	Deepseek-V3	0.1187	0.1581	0.1356
	Fino1-8B	0.0336	0.0140	0.0197
Extreme Classification	GPT-4o	0	0	0
	Deepseek-V3	0	0	0
	Fino1-8B	0	0	0

Table 8 shows that extreme classification is not a valid way to evaluate LLMs on XBRL tagging. This single-step formulation requires presenting the full space of thousands of US-GAAP concepts and directly assigning tags to raw text, without first identifying which spans are true numerical entities. In practice, this is infeasible: the model cannot realistically process the entire label space and is forced to rely only on its internal knowledge, which leads to meaningless zero-shot predictions. As the results indicate, all

Table 9: Accuracy comparison across the full pipeline, FinCL subtask, and rerank-only settings.

Model	Pipeline	FinCL	Rerank-only
GPT-4o	0.1052	0.1829	0.2023
Deepseek-V3	0.1074	0.1889	0.2144
Llama-4-Scout-17B-16E-Instruct	0.1045	0.1649	0.1792
Llama-3.3-70B-Instruct	0.0665	0.1318	0.1482
DeepSeek-R1-Distill-Qwen-32B	0.0814	0.1141	0.1258
Qwen3-32B	0.1127	0.1277	0.1402
gemma-2-27b-it	0.0533	0.1099	0.1210
Qwen3-14B	0.1069	0.1144	0.1321
Llama-3.1-8B-Instruct	0.0575	0.0913	0.1076
Llama-3.2-3B-Instruct	0.0178	0.0415	0.0566
Qwen3-1.7B	0.0598	0.0735	0.0880
Qwen3-0.6B	0.0362	0.0414	0.0533
Fino1-8B	0.0419	0.0807	0.0908

tested models, including GPT-4o and DeepSeek-V3, collapse to zero precision, recall, and F1 under this setting.

In contrast, our `FINTAGGING` benchmark reformulates the task into a two-stage process that aligns with how tagging is actually performed. The FinNI first detects and types coarse-grained numerical entities from text and tables, ensuring that only valid values enter the linking step. The FinCL then narrows the taxonomy search space by retrieving a candidate set of relevant tags and requiring the LLM to select the best match using contextual reasoning. With this design, models achieve non-trivial performance: for example, DeepSeek-V3 reaches an F1 of 0.1356 and GPT-4o 0.1026. These results demonstrate that FinTagging provides a more faithful evaluation of LLMs’ zero-shot tagging capabilities, surfacing meaningful signals where extreme classification fails.

4.4 Error propagation in `FINTAGGING` pipeline

Table 9 breaks down model accuracy across three evaluation settings: full pipeline, FinCL, and reranking-only in FinCL with guaranteed gold concepts in the top 200 candidates. The results reveal clear evidence of cascading errors. End-to-end pipeline accuracy is consistently the lowest, since errors from entity extraction (FinNI) and retrieval both propagate into concept linking. When gold entities are provided, accuracy improves substantially (e.g., GPT-4o from 0.1052 to 0.1829; DeepSeek-V3 from 0.1074 to 0.1889), confirming that entity identification is a key bottleneck in this context. With gold candidates guaranteed in the pool (rerank-only), performance further improves (e.g., GPT-4o 0.2023; DeepSeek-V3 0.2144), showing that narrowing the label space alleviates but does not eliminate difficulty, as semantic ambiguity among fine-grained US-GAAP concepts remains challenging.

4.5 Error cases

To better illustrate the semantic ambiguity challenge, we present a representative error case from our best-performing model, DeepSeek-V3. Consider the following input paragraph:

“Cash equivalents include term deposits with banks, money market funds, and all highly liquid investments with original maturities of three months or less..... At December 28, 2024, we had restricted cash of \$31 million recorded in other current assets and restricted cash of \$121 million recorded in other non-current assets.”

Table 10: Error cases from DeepSeek-V3 rerank.

Case 1: \$31m	
Gold Concept	us-gaap:RestrictedCashAndCashEquivalentsAtCarryingValue
Gold Concept Rank	18 / 200 (not in Top-5)
Model Prediction	us-gaap:RestrictedCashAndCashEquivalentsCurrent
Case 2: \$121m	
Gold Concept	us-gaap:RestrictedCashAndCashEquivalentsNoncurrent
Gold Concept Rank	2 / 200 (in Top-5)
Model Prediction	us-gaap:CashCashEquivalentsRestrictedCashAndRestrictedCashEquivalents

This paragraph contains 2 “monetary entities”: 31 and 121. The gold concepts and predicted concepts that are assigned to both entities are shown in Table 10. For case 1, the model’s choice was likely influenced by the phrase “recorded in current assets”. Yet the paragraph describes the overall composition of restricted cash, and the correct concept reflects this aggregate view rather than a single classification. For case 2, although both concepts denote non-current restricted cash, the model selected a narrower variant, showing the difficulty of distinguishing between highly similar concepts with subtle differences in scope.

5 Conclusion

This paper presents FINTAGGING, a benchmark for evaluating large language models on XBRL tagging of real-world financial reports. The task is divided into two subtasks, financial numeric identification (FinNI) and concept linking (FinCL), to enable fine-grained evaluation of both information extraction and taxonomy alignment. Results show that while LLMs generalize well to long-tail entities and perform competitively in zero-shot settings, they struggle with accurate alignment to GAAP concepts. This reveals limitations in structure-aware reasoning and highlights the need for better semantic understanding. FINTAGGING offers a foundation for advancing research in XBRL tagging and regulatory reporting. The limitations of our work and directions for future research are discussed in Appendix A.

References

- [1] Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063* (2019).
- [2] Matthew Bovee, Alexander Kogan, Kay Nelson, Rajendra P Srivastava, and Miklos A Vasarhelyi. 2005. Financial reporting and auditing agent with net knowledge (FRAANK) and extensible business reporting language (XBRL). *Journal of Information Systems* 19, 1 (2005), 19–41.
- [3] Zhiyuan Cao, Vipina K Keloth, Qianqian Xie, Lingfei Qian, Yuntian Liu, Yan Wang, Rui Shi, Weipeng Zhou, Gui Yang, Jeffrey Zhang, et al. 2025. The Development Landscape of Large Language Models for Biomedical Applications. *Annual Review of Biomedical Data Science* 8 (2025).
- [4] Zhiyu Chen, Wenhui Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. 2021. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122* (2021).
- [5] Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *arXiv preprint arXiv:2210.03849* (2022).
- [6] DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv:2501.12948 [cs.CL]* <https://arxiv.org/abs/2501.12948>
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR abs/1810.04805* (2018). *arXiv:1810.04805* <http://arxiv.org/abs/1810.04805>
- [8] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muenninghoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation. doi:10.5281/zenodo.12608602
- [9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Alkil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
- [10] Shijie Han, Haoqiang Kang, Bo Jin, Xiao-Yang Liu, and Steve Yang. 2024. XBRL-Agent: Leveraging Large Language Models for Financial Report Analysis. In *Proceedings of the 5th ACM International Conference on AI in Finance (ICAIF ’24)*.
- [11] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).
- [12] Rik Koncel-Kedziorski, Michael Krundick, Viet Lai, Varshini Reddy, Charles Lovering, and Chris Tanner. 2023. Bizbench: A quantitative reasoning benchmark for business and finance. *arXiv preprint arXiv:2311.06602* (2023).
- [13] Aixin Liu, Bei Feng, Bing Xue, Binxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).
- [14] Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ioannis Androulopoulos, and Georgios Palioras. 2022. FiNER: Financial numeric entity recognition for XBRL tagging. *arXiv preprint arXiv:2203.06482* (2022).
- [15] Chenxi Ma, Chen Huang, Jiaxin Wei, and Xu Sun. 2022. Label Semantics Enhanced Financial Entity Recognition. *arXiv preprint arXiv:2203.06482* (2022).
- [16] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2200–2209.
- [17] AI Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. [https://ai.meta.com/blog/llama-4-multimodal-intelligence/_checked_on_4_7_\(2025\).2025](https://ai.meta.com/blog/llama-4-multimodal-intelligence/_checked_on_4_7_(2025).2025).
- [18] Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Srivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarsi Ghosh, et al. 2022. Ecsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. *arXiv preprint arXiv:2210.12467* (2022).
- [19] Xueqing Peng, Triantafyllos Papadopoulos, Efstrathia Soufleri, Polydoros Giannouris, Ruoyu Xiang, Yan Wang, Lingfei Qian, Jimin Huang, Qianqian Xie, and Sophia Ananiadou. 2025. Plutus: Benchmarking Large Language Models in Low-Resource Greek Finance. *arXiv:2502.18772 [cs.CL]*
- [20] Lingfei Qian, Weipeng Zhou, Yan Wang, Xueqing Peng, Jimin Huang, and Qianqian Xie. 2025. Fino1: On the Transferability of Reasoning Enhanced LLMs to Finance. *arXiv preprint arXiv:2502.08127* (2025).
- [21] Jim Richards, Barry Smith, and Ali Saeedi. 2006. An introduction to XBRL. Available at SSRN 1007570 (2006).
- [22] Rachit Saini, Ankit Gupta, and Harshil Singh. 2021. GalaXC: Graph Neural Networks for Extreme Classification in Financial Text. *arXiv preprint arXiv:2104.05709* (2021).
- [23] Agam Shah, Ruchit Vithani, Abhinav Gullapalli, and Sudheer Chava. 2023. Finer: Financial named entity recognition dataset and weak-supervision model. *arXiv preprint arXiv:2302.11157* (2023).
- [24] Soumya Sharma, Subhendu Khatuya, Manjunath Hegde, Afreen Shaikh, Koustuv Dasgupta, Pawan Goyal, and Niloy Ganguly. 2023. Financial numeric extreme labelling: A dataset and benchmarking. In *Findings of the Association for Computational Linguistics: ACL 2023*. 3550–3561.
- [25] Soumya Sharma, Tapas Nayak, Arusarka Bose, Ajay Kumar Meena, Koustuv Dasgupta, Niloy Ganguly, and Pawan Goyal. 2022. FinRED: A dataset for relation extraction in financial domain. In *Companion Proceedings of the Web Conference 2022*. 595–597.
- [26] Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information processing & management* 45, 4 (2009), 427–437.
- [27] Gemma Team. 2024. Gemma. (2024). doi:10.34740/KAGGLE/M/3301
- [28] Qwen Team. 2025. Qwen3 Technical Report. *arXiv:2505.09388 [cs.CL]* <https://arxiv.org/abs/2505.09388>
- [29] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546* (2019).
- [30] Richard Zhe Wang. 2023. Standardizing XBRL financial reporting tags with natural language processing. Available at SSRN 4613085 (2023).
- [31] Yan Wang, Lingfei Qian, Xueqing Peng, Jimin Huang, and Dongji Feng. 2025. OrdRankBen: A Novel Ranking Benchmark for Ordinal Relevance in NLP. *arXiv:2503.00674 [cs.IR]*
- [32] Yan Wang, Jian Wang, Huiyi Lu, Bing Xu, Yijia Zhang, Santosh Kumar Banbhrauni, Hongfei Lin, et al. 2022. Conditional probability joint extraction of nested biomedical events: design of a unified extraction framework based on neural networks. *JMIR Medical Informatics* 10, 6 (2022), e37804.

- [33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
- [34] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814* (2019).
- [35] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *arXiv preprint arXiv:2306.05443* (2023).
- [36] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624* (2021).

A Limitations

In this work, we propose FINTAGGING and conduct a comprehensive analysis of LLM performance on the task. However, several limitations remain. First, we have not yet evaluated the latest models, such as GPT-5, gpt-oss, and Qwen3-235B. We are currently conducting these experiments and will report the results in future work. Second, we collected only a small dataset for fine-tuning PLMs and did not construct task-specific training data for LLMs. Future work may explore building LLM-based financial tagging agents through targeted fine-tuning to further enhance performance.

B Significance analysis

To further examine the comparative performance across models, we conducted a pairwise significance analysis, with the results summarized in Figure 4. The upper-triangular matrix in SubFigure 4a presents the pairwise significance outcomes, where a value of 1 indicates that the row model’s performance is significantly different from the column model. In contrast, 0 indicates that the difference is not statistically significant. The diagonal elements are set to zero, since self-comparisons are not meaningful.

As shown in SubFigure 4b, the number of significant pairwise differences varies across models, reflecting heterogeneous performance behaviors on the evaluated benchmark. The models DeepSeek-R1-Distill-Qwen-32B and Qwen3-32B exhibit the largest numbers of significant differences, followed by GPT-4o, Llama-4, and Qwen3-14B, suggesting that their performance diverges more noticeably from other models. These results may indicate that models within the Qwen family display stronger variability across comparisons, potentially due to differences in training objectives or architectural configurations. In contrast, Deepseek-V3, Llama-3.3-70B-Instruct, gemma-2-27b-it, and Fino1-8B demonstrate relatively fewer significant differences, implying that their performance remains more stable and comparable to other high-performing models. Overall, these findings suggest that larger or instruction-tuned models tend to produce more consistent outcomes, while models with smaller sizes or domain-oriented tuning show greater variation and clearer statistical separations under significance testing.

Overall, this analysis highlights not only which models outperform or underperform others but also which models behave similarly when subjected to statistical evaluation. The pairwise significance matrix, therefore, provides an interpretable view of model robustness and performance distinctiveness beyond average accuracy metrics.

C Literature Review

The XBRL provides a comprehensive taxonomy for financial reporting, encompassing thousands of detailed tags corresponding to concepts within financial statements. Applying NER to assign XBRL tags is an emerging yet challenging area.

C.1 XBRL Tagging benchmark

FiNER systematically benchmarked several neural architectures on the finer-139 dataset to address numeric-heavy XBRL tagging [14]. The initial experiments showed that standard BERT underperforms

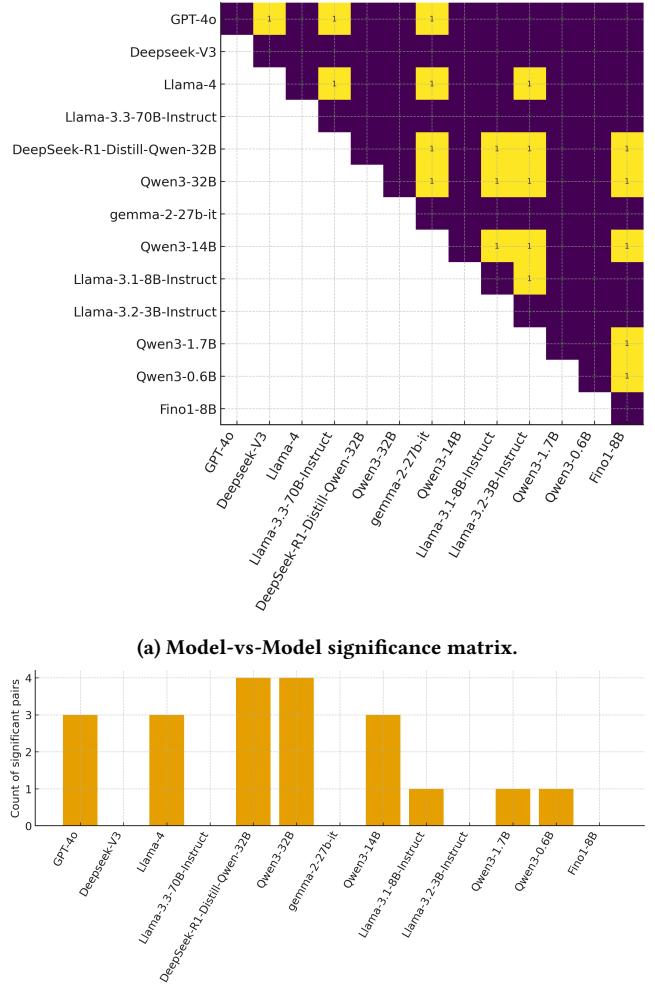


Figure 4: Visualization of significance analysis across models.
(a) shows the pairwise-bootstrap significance matrix, while
(b) summarizes how many significant differences each model has relative to others.

due to subword fragmentation, then the authors introduced pseudo-token strategies replacing numerals with [NUM] or [SHAPE] tokens to stabilize label assignment across fragmented numeric spans. These strategies, combined with domain-specific pretraining on SEC-BERT, significantly improved tagging performance, reaching 82.1 micro-F1 without the need for computationally expensive CRF layers. Their experiments demonstrated that subword-aware models with numeric-aware pseudo-tokens outperform both word-level BiLSTMs and vanilla BERT, particularly in numeric-heavy contexts, and avoid nonsensical label sequences. FNXL extended this benchmarking paradigm to a much larger label space of 2,794 US-GAAP tags, reframing the task as an extreme classification problem [24]. They compared the FiNER sequence labeling approach with a two-step pipeline that first identifies numeric spans and then assigns

labels using AttentionXML. While FiNER achieved stronger micro-F1 (75.84), reflecting better performance on frequent tags, AttentionXML outperformed FiNER in macro-F1 (47.54), highlighting its strength in predicting infrequent, tail-end labels. FNXL further evaluated both models under a Hits@k setting, confirming that label recommendations from the AttentionXML pipeline could substantially reduce manual effort and maintain high inter-annotator agreement. Together, these benchmarks reveal the need for context-aware reasoning and label-ranking mechanisms in realistic XBRL tagging scenarios.

C.2 XBRL Tagging Methods

The previous studies also explored the approaches address scalability, semantic ambiguity, and reasoning gaps in XBRL tagging to improve the performance. Saini et al. [22] proposed GalaXC is a graph-based extreme classification framework that jointly learns over document-label graphs with per-label attention across multi-hop neighborhoods. By integrating label metadata and transitive label correlations, GalaXC outperformed leading deep classifiers by up to 18% in micro-F1 on standard benchmarks and achieved 25% gains in warm-start scenarios where partial labels are available. Moreover, Wang et al. [30] addressed the practical challenge of custom tag standardization through a semantic similarity pipeline that leverages TF-IDF, Word2Vec, and FinBERT embeddings. Although unsupervised, the method was tested across nearly 200,000 custom tags from SEC filings between 2009 and 2022, and showed strong alignment performance, with vector-based mappings identifying viable standard tag candidates for a substantial proportion of non-compliant elements—offering a low-cost, interpretable solution for downstream financial analysis. Shifting focus from classification to comprehension, XBRL-Agent evaluated the capabilities of large language models to reason over full XBRL reports [10]. The authors introduced two task types—domain taxonomy understanding and numeric reasoning and found that base LLMs often hallucinated or misinterpreted financial content. To overcome these issues, XBRL-Agent incorporated retrieval-augmented generation (RAG) and symbolic calculators within an LLM-agent framework. The enhanced system achieved a 17% accuracy gain on domain query tasks and a 42% boost on numeric reasoning queries compared to base LLMs, validating the utility of modular tool augmentation. These improvements enabled reliable multi-step reasoning over complex disclosures such as debt instruments and derivative gains, which are difficult to capture using span-level classifiers. Collectively, these works broaden the methodological landscape of XBRL tagging from graph-based label propagation and embedding-based normalization to LLM-driven report analysis and point to a hybrid future where structured priors and reasoning tools jointly support accurate, scalable financial information extraction.

C.3 Financial Evaluation Benchmarks

In parallel to XBRL-specific advances, the financial NLP community has developed comprehensive benchmarks to assess broader capabilities in information extraction, numerical reasoning, and document understanding. FiNER-ORD [23] introduced a high-quality, domain-specific NER dataset annotated over financial news, emphasizing general entity types like persons, organizations, and locations.

While not numerically focused like FiNER-139, it highlights the lexical diversity of financial discourse and establishes a strong baseline for testing pretrained and zero-shot LLMs in real-world financial NER scenarios. FinQA [4] pushed toward explainable QA by pairing expert-written questions with annotated multi-step reasoning programs derived from earnings reports. ConvFinQA [5] extended this challenge to conversational contexts, simulating real-world question flows over sequential financial queries. TAT-QA [36] focused on hybrid tabular-text reasoning and required models to align cell values and document narratives, often involving aggregation, comparison, and unit-scale interpretation. Pixiu [35] introduced a broader evaluation framework by releasing FinMA, a financial LLM instruction-tuned across five tasks, and assessing it on a new benchmark covering sentiment classification, QA, summarization, NER, and stock prediction. BizBench [12] framed financial QA as program synthesis over realistic, multi-modal contexts, integrating reasoning, code generation, and domain knowledge into a single evaluation pyramid. While these benchmarks highlight the growing ability of models to integrate structured and unstructured financial data, they overlook taxonomy-driven fact alignment and do not support the structured output formats required for XBRL tagging.

D Data and Ticker Information

D.1 Overall Data Annotation

Different from the filtered dataset described in Section 3.3, we construct an overall annotation dataset based directly on the unfiltered raw data. As summarized in Table 11, the dataset consists of both sentence-level and table-level instances drawn from the narrative and tabular content of financial reports. In total, we annotated 15,986 sentence instances and 12,801 table instances, including 7,768 positive and 8,218 negative sentences, as well as 8,709 positive and 4,092 negative tables. On average, textual sentences contain 80.91 tokens, while table sequences are considerably longer, averaging 1,212.42 tokens. Across all data, we identified 261,457 numerical entities linked to 3,953 unique US-GAAP concepts.

Table 11: Statistical information for the original dataset in our benchmark. Tokens are calculated using the “cl100k_base” tokenizer (\pm standard deviation).

Item	Sentence	Table
Positive instances	7,768	8,709
Negative instances	8,218	4,092
Avg. Tokens/S	80.91 ± 63.62	1212.42 ± 1421.76
Avg. Entities/S	1.24 ± 1.82	18.87 ± 37.16
Avg. Concepts/S	1.24 ± 1.82	18.87 ± 37.16
Total Entities		261,457
Unique Concepts		3,953

D.2 The Statistics of the Tickers

Figure 5, Figure 6, and Figure 7 illustrate the distribution of 142 tickers across industry sectors, market capitalization categories, and geographic regions, highlighting the diversity of our collected financial reports. Considering that practical XBRL tagging practices often vary by industry, company size, and legal jurisdiction, we

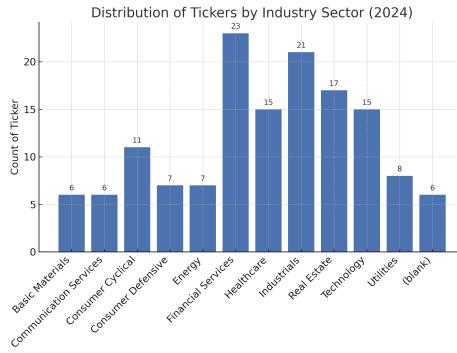


Figure 5: Distribution of Tickers by Industry Sector.

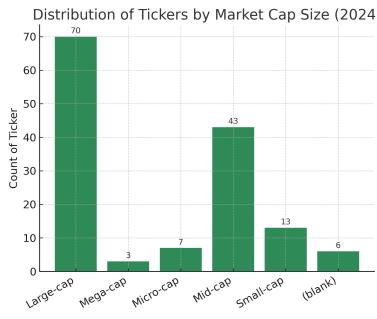


Figure 6: Distribution of Tickers by Market Cap Size.

curated a diverse set of companies covering all 11 major industry sectors. The sample maintains a balanced distribution across firm sizes, following market capitalization categories: micro-cap (<\$300M), small-cap (\$300M–\$2B), mid-cap (\$2B–\$10B), large-cap (\$10B–\$200B), and mega-cap (>\$200B). In addition, we incorporated firms from over 30 states and international jurisdictions to capture regional differences in reporting and tagging conventions. This diversity ensures that the benchmark reflects realistic heterogeneity observed in financial disclosures across sectors, scales, and regulatory environments.

E Benchmark Granularity Justification

Each 10-K report follows the SEC’s XBRL filing standard, where all narrative and tabular disclosures are linked through machine-readable tags defined in the US-GAAP taxonomy. While the XBRL framework provides document-level structure via instance, schema, and linkbase files, the tagging of financial values and concepts occurs at a much finer granularity—typically within individual sentences or table rows that contain numeric facts and their contextual descriptions.

For this reason, our benchmark adopts the **sentence** and **table sequence** as the fundamental input unit rather than entire reports. This design choice is motivated by three considerations:

- **Locality of Semantic Scope:** Financial facts and their US-GAAP tags are usually expressed within a single sentence or a bounded table region. Modeling at this level aligns the

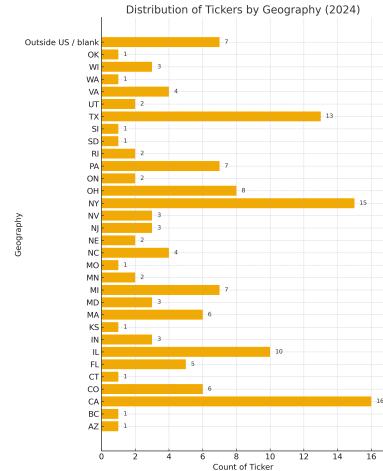


Figure 7: Distribution of Tickers by Geography.

input scope with the tagging scope, reducing noise from unrelated content within long filings.

- **Computational Efficiency:** Full 10-K filings often exceed hundreds of pages and contain heterogeneous sections (e.g., risk factors, management discussion, and financial statements). Processing the entire report as one input sequence would exceed model context limits and obscure local numerical relations. Segment-level inputs enable fine-grained supervision and efficient evaluation.
- **Faithful Tag Alignment:** Because each numerical entity is explicitly linked to its taxonomy tag in the XBRL instance document, using sentence or table segments ensures a one-to-one correspondence between input content and ground-truth annotations. This granularity preserves the original tagging fidelity of the filings.

Overall, this segmentation strategy allows our benchmark to remain consistent with how XBRL tagging operates in practice (on localized factual statements), while maintaining scalability and interpretability across large collections of filings.

F Validation Guideline

F.1 Task Definition

Given a context and an extracted triplet consisting of:

- an **entity**,
- its **entity type**,
- a **US-GAAP concept**,

Your task is to determine whether the triplet is correct.

Make a binary decision:

- 1 for correct,
- 0 for incorrect.

F.2 Validation Procedure

Follow the rules below, the definitions of entity type are shown in Table 12:

- (1) **Entity Check** Determine whether the extracted entity is a numerical value.
 - If yes, proceed to Step 2.
 - If no, set `is_correct` = 0.
- (2) **Entity Type Validation** Verify whether the identified entity type is correct based on the context and the definitions in Table 12.
 - If yes, proceed to Step 3.
 - If no, set `is_correct` = 0.
- (3) **US-GAAP Concept Validation** Assess whether the assigned US-GAAP concept is appropriate based on the taxonomy tag definitions.
 - If yes, set `is_correct` = 1.
 - If no, set `is_correct` = 0.

Table 12: Entity type definitions used for validation.

Entity Type	Definition
monetaryItemType	Financial amounts expressed in currency, such as revenue, profit, or total assets.
integerItemType	Counts of discrete items, such as the number of employees or total transactions.
perShareItemType	Per-share values, such as earnings per share (EPS) or book value per share.
sharesItemType	Counts of shares, such as outstanding shares or ownership stakes.
percentItemType	Ratios or percentages, such as tax rates, growth rates, or discount rates, usually expressed with a percentage symbol (%).

F.3 Important Instructions

- (1) **Non-Arabic Formats** Financial numerical entities may appear in word form (e.g., *ten million*) and must be correctly identified and converted into standard numerical format.
- (2) **Magnitude Terms** If a number is followed by a magnitude term (e.g., *hundred, million, billion*), do not expand it into the full numerical value:
 - *Two hundred* → extract only *two*, not *200*.
 - *10.6 million* → extract only *10.6*, not *10,600,000*.
- (3) **Standardization of Format** Remove formatting symbols while preserving the numerical value:
 - Remove currency symbols (e.g., *USD*).
 - Remove percentage signs (e.g., *%*).
 - Remove commas (e.g., *1,000* → *1000*).

G Evaluation Metrics

To provide a fair evaluation of overall benchmark performance, we adopt a set of metrics, focusing primarily on macro-level and micro-level evaluation strategies inspired by the previous work [24]. **Macro-level** evaluation computes precision, recall, and F1 scores independently for each BIO-concept label derived from the US-GAAP taxonomy, and then averages them without weighting. This ensures that each concept, including rare or infrequent ones, contributes equally to the final score, making it especially suitable for domains with skewed label distributions. In contrast, **micro-level** evaluation aggregates token-level true positives, false positives, and false negatives across all labels before computing precision, recall, and F1. This approach emphasizes the model’s overall tagging accuracy by treating every token equally and thus better reflects performance on frequent concepts. Together, these two metrics provide a balanced view of both per-concept performance and overall tagging quality.

For the FinNI subtask, the objective is to extract correct (**entity**, **type**) pairs, that is (Fact, Type), from the financial document. Let $\mathcal{G} = (e_i, l_i)$ denote the set of ground-truth (entity, type) pairs, and $\mathcal{P} = (e'_i, l'_i)$ denote the set of predicted (entity, type) pairs. We evaluate the performance based on the following metrics:

$$\text{Precision} = \frac{|\mathcal{G} \cap \mathcal{P}|}{|\mathcal{P}|} \quad (5)$$

$$\text{Recall} = \frac{|\mathcal{G} \cap \mathcal{P}|}{|\mathcal{G}|} \quad (6)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

where $(e_i, l_i) = (e'_j, l'_j)$ if and only if both the entity span e and its assigned type l exactly match.

For the FinCL subtask, Given a set of queries $Q = \{q_1, q_2, \dots, q_N\}$, where each q_i is associated with a ground-truth concept c_i^* and a predicted concept \hat{c}_i , the accuracy is defined as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \delta(\hat{c}_i, c_i^*) \quad (8)$$

where $\delta(\hat{c}_i, c_i^*) = 1$ if $\hat{c}_i = c_i^*$, and 0 otherwise.

H Evaluation Models Details

Table 13 provides an overview of the models evaluated in this study, categorized by openness, domain specialization, and architectural foundation. The evaluation covers a diverse range of models:

- Closed-source LLMs: We include GPT-4o [11], accessed via OpenAI’s API, as a representative of cutting-edge proprietary models with demonstrated performance across a variety of NLP tasks. Although model size details are undisclosed, GPT-4o serves as an upper-bound reference in our benchmark.
- Open-source LLMs: This group encompasses recent, high-performing open models such as DeepSeek-V3 (685B) [13], DeepSeek-R1-Distill-Qwen (32B) [6], and multiple variants of Qwen3 [28] (ranging from 0.6B to 32B). We also include Llama-3.3, Llama-3.2, and 3.1 variants [9] (70B, 3B, and 8B), Llama-4-Scout-17B-16E-Instruct [17] (109B), as well as Google’s Gemma-2-27B [27], to ensure architectural diversity and scalability comparison. These models are primarily instruction-tuned and optimized for general-purpose NLP tasks.
- Financial-specific LLMs: We evaluate Fino1-8B [20], a domain-specialized model trained on financial corpora, designed to better capture the terminology and structure unique to financial disclosures. This category allows us to assess the benefits of domain adaptation in complex tagging and reasoning tasks.
- Pretrained Language Models (PLMs): To establish strong baselines, we include non-generative encoder models: BERT-large [7], FinBERT [1], and SECBERT [14]. These models have been widely used in prior financial NLP tasks and allow for a comparative analysis between generative LLMs and traditional pretrained models in terms of domain understanding and structured output capability.

Together, these models offer a comprehensive evaluation spectrum, from general-purpose to domain-specific, encoder-based to

decoder-based, and open to closed source, facilitating an in-depth assessment of their performance across our proposed benchmark tasks.

Table 13: Model categories and corresponding repositories.

Model	Size	Source
Closed-source Large Language Models		
GPT-4o	-	gpt-4o-2024-08-06
Open-source Large Language Models		
DeepSeek-V3	685B	deepseek-ai/DeepSeek-V3
Llama-4-Scout-17B-16E-Instruct	10PB	meta-llama/Llama-4-Scout-17B-16E-Instruct
Llama-3.3-70B-Instruct	70B	meta-llama/Llama-3.3-70B-Instruct
DeepSeek-R1-Distill-Qwen	32B	deepseek-ai/DeepSeek-R1-Distill-Qwen-32B
Qwen3-32B	32B	Qwen/Qwen3-32B
gemma-2-27b-it	27B	google/gemma-2-27b-it
Qwen3-14B	14B	Qwen/Qwen3-14B
Llama-3.1-8B-Instruct	8B	meta-llama/Llama-3.1-8B-Instruct
Llama-3.2-3B-Instruct	3B	meta-llama/Llama-3.2-3B-Instruct
Qwen3-1.7B	1.7B	Qwen/Qwen3-1.7B
Qwen3-0.6B	0.6B	Qwen/Qwen3-0.6B
Financial-specific Large Language Models		
Fino1	8B	TheFinAI/Fino1-8B
Pretrained Language Models		
BERT-large	~340M	google/bert/bert-large-uncased
FinBERT	~110M	ProsusAI/finbert
SECBERT	~110M	nlpaeub/sec-bert-base

I The details for the fine-tuning PTMs

I.1 Training data collection and processing

Similar to the collection process for the FinTAGGING benchmark data, we gathered an additional 10 annual 10-K financial reports filed with the SEC for the period from February 13, 2024, to February 13, 2025, as summarized in Table 14. These reports contain a total of 33,848 standard taxonomy-tagged facts. Using BeautifulSoup to parse these documents, we identified 22,847 narrative sentences (approximately 5.5 million characters) and 1,236 financial tables. The companies included in this dataset follow the XBRL standard, ensuring comprehensive coverage for training PTMs.

Table 14: Financial report statistics summary for raw training data

Item	Information
Report type	10-K
Period	2024-02-13 to 2025-02-13
#Company	10
#Sentence	22,847
#Table	1,236
#Characters	5,539,198
#Standard Tags	33,848

After collection, we employed the same procedure to filter texts and tables, subsequently annotating numerical entities, entity types, and US-GAAP tags (concepts). Finally, as detailed in Table 15, we generated a total of 1,116 sentences and 953 tables as the training set for PTMs. Specifically, the sentence-level data consists of 558 positive and 558 negative instances, averaging approximately 84.24 tokens (± 69.29), with 1.22 annotated entities and concepts per sentence. The table-level data comprises 594 positive and 359 negative instances, with a significantly higher average of 1,281.86 tokens ($\pm 6,438.37$), and approximately 25 entities and concepts annotated per table. Overall, the annotated dataset includes 25,199 entities, covering 1,435 unique US-GAAP concepts.

Table 15: Statistics of training data (tokens calculated with “cl100k_base” tokenizer, \pm standard deviation).

Structure	Pos/Neg	#Instance	Avg. Tokens/S	Avg. Entities/S	Avg. Concepts/S	Total Entities	Unique Concepts		
Sentence	Positive	558	84.24 \pm 69.29	1.22 \pm 1.78	1.22 \pm 1.78	25,199	1,435		
	Negative	558							
Table	Positive	594	1281.86 \pm 6438.37	25.00 \pm 213.77	25.00 \pm 213.77				
	Negative	359							

However, to align with the extreme classification format used in previous XBRL tagging benchmarks, we directly adopt the US-GAAP tags as entity labels, annotating each token in sentences and tables using the BIO scheme. Specifically, B denotes the beginning of an entity phrase, I marks the continuation (inside) of an entity phrase, and O indicates tokens outside of any entity. As shown in Figure 8, “4.9” and “4.5” are single-token numerical entities labeled only with a B prefix (e.g., “B-us-gaap:AccountsReivableNetNoncurrent”). To comprehensively cover all US-GAAP tags, we combine the entire set of 17,388 tags from the US-GAAP 2024 taxonomy with the BIO labeling scheme to construct an extreme classification label space, resulting in 34,777 unique entity labels ($2 \times 17388 + 1$).

After constructing the training set, we reconstruct the testing set from the original benchmark dataset. The training settings are detailed below.

I.2 Training settings

We fine-tune three pretrained models, BERT-large [7], FinBERT [1], and SECBERT [14], on our training set using the HuggingFace Transformers library. All models are trained with a batch size of 4, a learning rate of 3e-5, and for 20 epochs. Optimization is performed using AdamW without gradient accumulation or early stopping. Token classification heads are randomly initialized and trained jointly with the base encoder. Input sequences are tokenized with a maximum length of 512, and labels are aligned at the sub-token level following the BIO tagging scheme. Loss is computed only on the first sub-token of each word to avoid misalignment bias.

Training is conducted on two NVIDIA A5000 GPUs (24GB each) using data parallelism for 24 hours. All other hyperparameters follow the default settings in the HuggingFace Trainer API. Models are evaluated using the checkpoint from the final epoch. All experiments are run under a fixed random seed to ensure reproducibility.

J Retrieval Results in FinCL subtask

We investigate the impact of different context construction strategies for the queried entity at the **retrieval stage**. We consider two approaches: Fixed-Window Context (FWC) and Structure-Aware Context (SAC). In the FWC strategy, context is constructed by extracting a fixed window of 50 characters before and after the entity mention, regardless of whether the entity appears in a sentence or a table. In contrast, the SAC strategy builds context based on the structural location of the entity: if the entity appears in a sentence, the entire sentence is used; if the entity appears in a table, we linearize the entire row into a Markdown-style key-value format to serve as the context.

From Table 16, we observe that the SAC strategy consistently outperforms FWC, particularly in the table context, highlighting the importance of aligning the context window with the underlying structural unit.

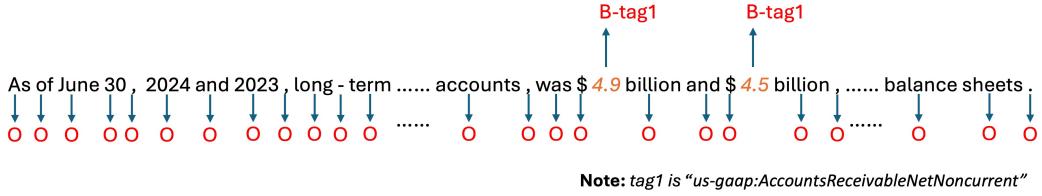


Figure 8: An example for training set annotation with BIO scheme.

Table 16: Acc@k retrieval performance on the FinCL task.

FWC: fixed window-based; SAC: structure-aware.

Strategy	Structure	Acc@1	Acc@10	Acc@20	Acc@50	Acc@100	Acc@150	Acc@200
FWC	Sentence	0.0658	0.2614	0.3562	0.4650	0.5618	0.6114	0.6502
	Table	0.0000	0.0029	0.0038	0.0042	0.0050	0.0065	0.0089
	Overall	0.0055	0.0237	0.0316	0.0409	0.0492	0.0544	0.0608
SAC	Sentence	0.0696	0.2742	0.3631	0.4798	0.5701	0.6125	0.6535
	Table	0.0159	0.0872	0.1245	0.1938	0.2452	0.2723	0.2959
	Overall	0.0202	0.1152	0.1534	0.2188	0.2727	0.3012	0.3274

For sentence-based entities, both strategies yield comparable results, with SAC achieving slightly higher Acc@1 (0.0696 vs. 0.0658) and showing marginal improvements across all cutoff values (e.g., Acc@100 of 0.5701 vs. 0.5618). This suggests that even for relatively unstructured text, preserving sentence boundaries provides minor benefits over fixed-length windows.

In contrast, for table-based entities, the performance gap is substantial. SAC achieves significantly higher retrieval accuracy (e.g., Acc@100 of 0.2452 vs. 0.0050), indicating that row-level context is far more informative than arbitrary character windows when dealing with tabular structures. FWC performs poorly in this setting, likely due to the fragmented and semantically sparse nature of partial table text.

When aggregating results across both structures, SAC outperforms FWC by a wide margin at all retrieval depths (e.g., Acc@200 of 0.3274 vs. 0.0608). These findings underscore the importance of structure-aware context construction, especially in scenarios where inputs span multiple formats such as sentences and tables.

Prompt Template for FinNI Subtask

You are a financial information extraction expert specializing in identifying financial numerical entities in XBRL reports.

Your task is to extract all such numerical entities from the provided text or serialized <table></table> data and classify them into one of five categories:

- "integerItemType": Counts of discrete items, such as the number of employees or total transactions.
- "monetaryItemType": Financial amounts expressed in currency, such as revenue, profit, or total assets.
- "perShareItemType": Per-share values, such as earnings per share (EPS) or book value per share.
- "sharesItemType": Counts of shares, such as outstanding shares or ownership stakes.
- "percentItemType": Ratios or percentages, such as tax rates, growth rates, or discount rates, usually expressed with a percentage symbol ("%").

Important Instructions:

- (1) Financial numerical entities are not limited to Arabic numerals (e.g., 10,000). They may also appear in word form (e.g., "ten million"), which must be correctly identified and converted into standard numerical format.
- (2) Not all numbers in the text should be extracted. Only those that belong to one of the five financial entity categories above should be included. Irrelevant numbers (such as phone numbers, dates, or general IDs) must be ignored.
- (3) If a number is followed by a magnitude term (e.g., Hundred, Thousand, Million, Billion), do not expand it into the full numerical value.
 - * "Two hundred" -> Extract only "two", not "200".
 - * "10.6 million" -> Extract only "10.6", not "10,600,000".
- (4) Standardize numerical formatting by removing currency symbols (e.g., "USD"), percentage signs ("%"), and commas (",") while preserving the numeric value. These elements must be removed to ensure consistency.
- (5) Output the extracted financial entities in JSON list format without explanations, structured as follows: {"result": [{"Fact": <Extracted Numerical Entity>, "Type": <Identified Entity Type>}]}]

Input: {text/table}

Output:

Figure 9: Prompt template used for the FinNI subtask.

Prompt Template for FinCL Subtask (Reranking)

You are a financial tagging assistant trained in US-GAAP taxonomy.

Given a query consisting of an entity, its type, its surrounding context, and the source format (either text or table), your task is to select the single most appropriate US-GAAP tag from a list of 200 candidate tags.

Make your decision by carefully analyzing the meaning and context of the entity and matching it with the semantics of the tags.

Only output one tag, the best match. Do not explain or list multiple tags. The output is a JSON format: {"result": <the best matched tag>}.

Input Query: <entity> + <entity type> + <context>
Candidate Tags: {Top 200 US-GAAP tags}

Answer:

Figure 10: Prompt template used for the Reranking stage in the FinCL subtask.

Prompt Template for Ablation

You are an XBRL tagging expert specializing in annotating financial numerical facts in XBRL reports.

Your task is to (1) extract all such numerical entities from the provided text or serialized <table></table> data, (2) classify them into one of five categories, and (3) assign an appropriate US-GAAP tag to each entity.

Categories:

- "integerItemType": Counts of discrete items, such as the number of employees or total transactions.
- "monetaryItemType": Financial amounts expressed in currency, such as revenue, profit, or total assets.
- "perShareItemType": Per-share values, such as earnings per share (EPS) or book value per share.
- "sharesItemType": Counts of shares, such as outstanding shares or ownership stakes.
- "percentItemType": Ratios or percentages, such as tax rates, growth rates, or discount rates, usually expressed with a percentage symbol ("%").

US-GAAP tags:

- A US-GAAP tag is a standardized semantic label used in XBRL filings to identify specific financial concepts defined by the U.S. Generally Accepted Accounting Principles (GAAP). Each tag represents a distinct accounting item and enables consistent, machine-readable financial reporting.
- Examples: "us-gaap:AssetsCurrentAbstract", "us-gaap:AccruedInsuranceNoncurrent".

Important Instructions:

- (1) Financial numerical entities are not limited to Arabic numerals (e.g., 10,000). They may also appear in word form (e.g., "ten million"), which must be correctly identified and converted into standard numerical format.
- (2) Not all numbers in the text should be extracted. Only those that belong to one of the five financial entity categories above should be included. Irrelevant numbers (such as phone numbers, dates, or general IDs) must be ignored.
- (3) If a number is followed by a magnitude term (e.g., Hundred, Thousand, Million, Billion), do not expand it into the full numerical value.
 - * "Two hundred" -> Extract only "two", not "200".
 - * "10.6 million" -> Extract only "10.6", not "10,600,000".
- (4) Standardize numerical formatting by removing currency symbols (e.g., "USD"), percentage signs ("%"), and commas (",") while preserving the numeric value. These elements must be removed to ensure consistency.
- (5) You should assign the most appropriate US-GAAP tag to each identified entity based on your internal understanding of the 2024 US-GAAP taxonomy.
- (6) Output the extracted financial entities in JSON list format without explanations, structured as follows: {"result": [{"Fact": <Extracted Numerical Entity>, "Type": <Identified Entity Type>, "Tag": <Assigned US-GAAP tag>}]}}

Input: {text/table}

Output:

Figure 11: Prompt template used for ablation study.