Financial Semantic Textual Similarity: A New Dataset and Model

Shanshan Yang, Steve Yang
School of Business
Stevens Institute of Technology
Hoboken, NJ, USA
{syang56, syang14}@stevens.edu

Feng Mai
Department of Business Analytics
University of Iowa
Iowa City, IA, USA
feng-mai@uiowa.edu

2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr) | 979-8-3503-5483-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CIFER62890.2024.10772793

***Abstract*—We introduce FinSTS, a novel dataset for financial semantic textual similarity (STS), comprising 4,000 sentence pairs from earnings calls and SEC filings. To improve models for the Financial STS task, we propose an active learning (AL) algorithm that efficiently selects informative sentence pairs for annotation by GPT-4 and creates high-quality training data. Using this approach, we train FinSentenceBERT, a model that generates semantic embeddings specifically for financial text. FinSentenceBERT establishes a new performance benchmark on FinSTS, outperforming models that use basic pooling strategies or are fine-tuned on general datasets. Surprisingly, a general SBERT model trained using our AL approach surpasses even models based on FinBERT, a language model pre-trained on financial text. Our research contributes a specialized dataset, model, and methodology that advance semantic understanding in the financial domain, with potential applications to other specialized domains.**

***Index Terms*—BERT, Representation learning, Active learning, Text similarity**

I. Introduction

Semantic Textual Similarity (STS) is a fundamental task in natural language understanding that assesses the degree of semantic equivalence between sentence pairs [1]. In the financial domain, STS techniques enable the semantic comparison and understanding of vast amounts of textual data, including regulatory filings, news articles, analyst reports, and social media content [2]. Recent studies have demonstrated the value of STS in various financial applications, for example, generating industry classifications based on product descriptions [3], quantifying corporate cultural values [4], and analyzing the effects of horizontal acquisitions on market power [5]. These diverse applications highlight the importance of accurate, scalable measurement of text similarity in financial texts to transform research and decision-making.

Traditional methods for computing financial text similarity often rely on lexical semantics based on bag-of-words representations [6] and fail to capture the semantic meaning and context of words. Advancements in pre-trained language models (PLMs), such as BERT [7], show promise in capturing semantic relationships between sentences by pre-training the encoder using masked language modeling and next sentence prediction objectives, which enable the model to learn contextual representations of words and sentences.

Researchers have adapted the BERT model to the financial domain by training the model from domain-specific corpora, resulting in models like FinBERT [8]. Compared to general-domain PLMs, FinBERT excels at text classification tasks, including tone classification and Environmental, Social, and Governance (ESG) classification. However, its capacity for STS remains unexplored. To measure the similarity between sentences, it may seem logical to directly compute the cosine similarity between the mean-pooled token embeddings generated by a BERT model. Yet this method often produces less accurate results than simpler techniques like averaging GloVe vectors [9]. To address these limitations, Reimers and Gurevych [9] propose Sentence-BERT (SBERT), a fine-tuned version of BERT using siamese networks to derive semantically meaningful sentence embeddings. SBERT significantly enhances PLMs' performance on STS tasks such as clustering and information retrieval. The success of SBERT in general-domain STS tasks highlights the potential for developing a similar approach tailored to the financial domain. Can we leverage the power of domain-adapted PLMs like FinBERT to generate semantically rich sentence representations for financial texts?

To address this research question, we aim to construct a financial STS (**FinSTS**[1]) dataset and develop a **FinSentenceBERT**[2] model. Constructing a high-quality dataset for FinSTS tasks is non-trivial due to the requirement of capturing gradations of meaning overlap between sentence pairs. Unlike binary classification, STS datasets need to reflect the degree of semantic similarity [1]. The examples are typically scored on an ordinal scale ranging from no meaning overlap (0) to complete semantic equivalence (5). Annotating financial sentence pairs requires careful consideration of both pragmatic and world knowledge. A diverse set of sentence pairs are also needed to represent the subtleties of semantic similarity in different contexts. In addition, we need to carefully consider which pairs of sentences to annotate. Annotating a large number of randomly selected sentence pairs could result in examples that are either highly similar or dissimilar, which provide little value for training and evaluation.

[1]https://huggingface.co/datasets/syang687/FinSTS
[2]https://huggingface.co/syang687/FinSentenceBERT

We propose a two-stage solution. In the first stage, which occurs *before* model training, we construct a large pool of unlabeled sentence pairs from financial reports and earnings calls. We then employ various sampling strategies to ensure a diverse representation of semantic patterns. We develop a financial STS gold set consisting of 4,000 sentence pairs. This gold set serves as a benchmark for model evaluation and testing.

In the second stage, we introduce an Active Learning (AL) algorithm to strategically select the most informative sentence pairs for annotation *during* model training. AL is a promising approach to address the challenges of dataset construction by focusing labeling the most valuable instances [10], [11]. While recent studies have demonstrated the value of AL for various classification tasks [10], the potential of AL combined with BERT for the STS task remains unexplored. Our AL algorithm integrates sentence pair selection with model training, iteratively selecting pairs that the model is most uncertain about and updating the training set accordingly. To efficiently label the selected pairs, we leverage the GPT-4 model, which offers high-quality annotations at a lower cost and time compared to traditional human labeling. We validate the quality of GPT-4 labels against human annotations on a subset of the gold set, which confirms its reliability for the AL process. This process results in a FinSentenceBERT model built upon the siamese network structure of the SBERT model [9].

Using the FinSTS dataset, we compare the performance of various embedding techniques in the financial domain, including mean-pooling of vanilla BERT/FinBERT, well-trained general-domain SBERT models, BERT/FinBERT models fine-tuned on the general-domain STS benchmark (STSb) dataset, as well as the improvement brought by AL. We find that: 1) mean pooling of vanilla FinBERT outperforms those of the general BERT models; 2) fine-tuning BERT and FinBERT on the STSb dataset improves upon the mean pooling approach and results in the fine-tuned BERT surpassing the fine-tuned FinBERT; and 3) our AL algorithm can further enhance model performance and narrow 95% confidence interval of model test performance, which suggests that the model is better-performed and more stable due to the AL. Surprisingly, the best-performing model is a general-domain SBERT model combined with AL, which outperforms the FinBERT model fine-tuned on STSb and then enhanced with AL.

The main contributions of this paper are as follows:

- We construct the FinSTS dataset, a benchmark dataset for financial semantic textual similarity.
- We develop a FinSentenceBERT model for generating semantically meaningful sentence embeddings in the financial domain.
- We introduce an AL algorithm to optimize sentence pair selection for annotation during STS model training, and demonstrate its value for the FinSTS task.
- We compare the performance of various sentence embedding models in finance.

Overall, the FinSTS dataset offers a new benchmark for evaluating financial semantic textual similarity, while the FinSentenceBERT model empowers finance researchers to explore novel applications and derive insights from financial text data. For example, it has the potential to enhance firm-based similarities across various dimensions, including product markets and innovation, and enables text-based measurement. These resources provide a foundation for future research and innovation in financial text analysis.

II. Related Work

Our work is primarily related to adapting and evaluating STS to various specialized domains, e.g., healthcare [12], [13], academic literature [14], and legal studies [15]. The accurate measurement of semantic similarity between text pairs is essential for tasks like information retrieval, summarization, and QA. The STS benchmark (STSb) dataset [1], a general-domain dataset constructed by selecting labeled English pairs from SemEval and SEM STS shared tasks, provides a foundation for comparable assessments across different research efforts and improved tracking of the state-of-the-art in semantic textual similarity. However, each domain presents unique challenges due to its specific terminology, complex concepts, and differences in semantics, which can hinder the effectiveness of general-purpose STS models.

Various domain-specific STS datasets, as listed in Table I, have been developed to address these challenges and facilitate research on domain-adapted STS models. For instance, the SPICED dataset [16] is a collection of scientific finding pairs annotated for information change to analyze scientific communication, including pairs from original papers and their corresponding news articles and tweets. The MedSTS dataset [12] is curated from clinical sentences de-identified from multiple patient records, with similarity scores computed using surface lexical similarity metrics. The CORD19-STS dataset [13] is generated from the CORD19 Open Research Dataset (CORD19) challenge, with sentence pairs scored for similarity using a SCI-BERT model fine-tuned with CORD-19 text. Despite the growing interest in domain-specific STS, there remains a paucity of datasets and models tailored to the financial domain, which is characterized by its unique language and semantic structures. The FinSTS dataset introduced in this paper aims to bridge this gap.

TABLE I
Examples of Domain-specific STS Research

Dataset	# Pairs	Domain	Annotators	Labels
SPICED [16]	6,000	Scientific Findings	Experts	Information Matching Scores (1-5)
CORD19-STS [13]	13,710	COVID-19	AMT users	Related, Somewhat-related, Not-related
MedSTS [12]	1,068	Clinical	Medical experts	Similarity Scores (0-5)
STSb [1]	8,628	General	AMT users	Similarity Scores (0-5)

While these domain-specific datasets have advanced STS research, general-purpose language models like BERT still face challenges in accurately capturing semantic similarity in

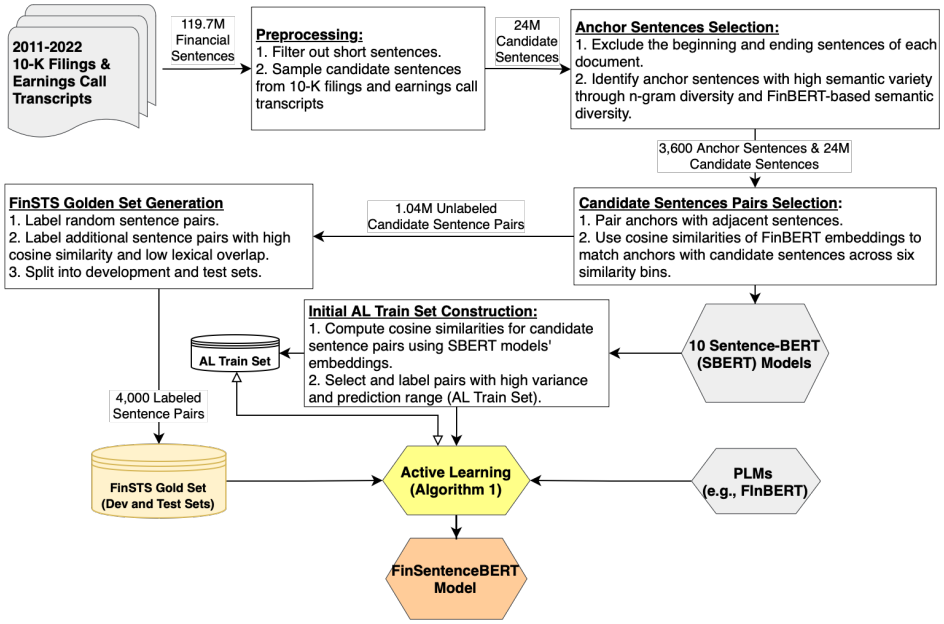

Fig. 1. Flow Chart of Analysis

specialized domains. PLMs, such as BERT [7], have significantly advanced the state-of-the-art in various NLP tasks. BERT learns contextual representations of words by pre-training on large corpora using masked language modeling and next sentence prediction objectives. However, BERT has limitations when generating well-performing embeddings for single sentences [9]. Common practices for converting BERT outputs to sentence embeddings, such as averaging contextual token embeddings or using the [CLS] token embedding, often yield suboptimal results compared to simpler methods like GloVe embeddings [17].

To address this limitation, several unsupervised methods have been proposed to tune PLMs for better sentence embeddings. Recent work include SimCSE [18], BERT-Whitening [19], and TSDAE [20]. These methods utilize contrastive learning, isotropy enhancement, and sequential denoising auto-encoders to improve sentence embeddings without the need for labeled data. Supervised learning methods, on the other hand, tend to achieve better results. Sentence-BERT (SBERT) [9] adapts the BERT model using siamese networks and fine-tunes it on the STS benchmark (STSb) dataset [1] and Natural Language Inference (NLI) datasets for improved sentence-level comparisons. This approach of training on annotated datasets leads to a large improvement in performance. However, a large amount of labeled training data is required, which can be expensive and time-consuming to obtain especially in domain-specific settings like finance. Striking a balance between supervised and unsupervised methods is crucial for developing effective domain-specific STS models. Active learning techniques strategically select the most informative instances for labeling. As such, they can potentially bridge this gap by reducing the annotation effort while maintaining high performance [10], [11].

Finally, our work is inspired by FinBERT [8], a finance domain-specific adaptation of BERT. FinBERT is developed using a large corpus of financial texts, including corporate annual and quarterly filings, financial analyst reports, and earnings conference call transcripts. This domain-specific pre-training allows FinBERT to better capture contextual information and incorporate finance knowledge compared to general-purpose language models. Huang et al. [8] demonstrate FinBERT's superior performance in sentiment classification tasks using analyst report sentences labeled by researchers as the primary benchmark. FinBERT achieves substantially higher out-of-sample accuracy than other popular approaches in finance and accounting research, such as dictionary-based, naïve Bayes, support vector machines, random forests, convolutional neural networks, and LSTMs. The advantage of FinBERT over other algorithms, including Google's original BERT model, is especially prominent when the training sample size is small and in texts containing financial words not frequently used in general texts. Additionally, FinBERT outperforms other models in identifying discussions related to ESG issues.

Despite the promising results of FinBERT in domain-specific classification tasks, its potential for STS tasks remains unknown. Combining the domain-specific pre-training of FinBERT with fine-tuning techniques like the ones for SBERT could potentially yield powerful models for domain-specific STS tasks in the financial domain. However, the effectiveness of this combination is not guaranteed and may depend on various factors such as the quality and quantity of the training data and the specific characteristics of the financial STS task.

III. Data and Methodology

In this section, we detail our approach to constructing the FinSTS dataset and developing the FinSentenceBERT model.

We propose a multi-stage approach (see Figure 1). The cornerstone of our method is an Active Learning (AL) algorithm that strategically selects the most informative sentence pairs for annotation, while iteratively refining the FinSentenceBERT model to capture subtle semantic differences in financial texts.

We prepare three key components before employing the AL algorithm. First, we compile a diverse pool of over 1 million unlabeled financial sentence pairs, sourced from earnings call transcripts and 10-K filings spanning 2011 to 2022. Second, we curate a gold FinSTS dataset of 4,000 labeled sentence pairs that serve as a benchmark for model evaluation. Third, we select a small initial training set of sentence pairs based on the uncertainty of predictions from an ensemble of general-domain Sentence-BERT models to kick-start the FinSentence-BERT model fine-tuning process. The following subsections provide a detailed description of each stage.

A. Financial Text Sources

We focus on two primary sources of financial texts: earnings call transcripts and three key sections of 10-K filings from 2011 to 2022. The selected 10-K sections include Item 1 (Business Description), Item 1A (Risk Factors), and Item 7 (Management's Discussion and Analysis). These sections are chosen for their informational value to investors [21]. By incorporating both formal writing and conversational language, our dataset captures a comprehensive view of financial discourse. We extract individual sentences from these sources using the NLTK sentence tokenizer [22], resulting in a corpus of approximately 119.7 million financial sentences from 10-K filings and earnings call transcripts.

The extracted 119.7 million financial sentences are preprocessed and filtered to include only those with more than five tokens, excluding sentences consisting of meaningless tokens such as page or line breakers. From this filtered set, 24 million sentences are randomly sampled, with each set of 1 million sentences drawn equally from the 12 years (2011-2022) and the two financial text sources. This approach balances computational efficiency with comprehensive coverage across time and text sources.

B. Construction of FinSTS Dataset

To capture the diversity and complexity of financial language, we construct the FinSTS dataset in a multi-step process. First, we select 3,600 anchor sentences to capture a wide range of semantic patterns and topics in financial texts. Then, we pair anchor sentences with selected sentences from the large corpus, yielding a pool of 1.044 million candidate sentence pairs. Finally, we generate the FinSTS Gold Set by labeling a subset of the candidate pairs, including randomly selected pairs and pairs with high semantic similarity but low lexical overlap, so that the dataset contains an even distribution of various degrees of semantic similarities. The following subsections discuss these steps in detail.

1) Anchor Sentences Selection: We select a total of 3,600 diverse anchor sentences from the 24 million sentences (150 anchor sentences from each set of 1 million sentences). To accommodate the strategy of pairing anchor sentences with their consecutive sentences, we exclude the beginning and ending sentences of each 10-K filing and earnings call transcript. The selection process involves iterating through each sentence in each set of 1 million sentences, computing the following two selection metrics for each sentence based on the already selected anchor sentences, and selecting the sentence as an anchor if both metrics meet their threshold conditions.

To ensure semantic diversity among the selected anchor sentences, we employ a weighted n-gram diversity metric, extending the diversity metric of the Density Weighted Diversity Ensemble (DWDS) [23]:

$$\beta(s_a, \mathcal{S}, n) = \frac{\sum_{x \in \text{n-gram}(s_a)} \mathbb{I}(x \notin \text{n-gram}(\mathcal{S}))}{|\text{n-gram}(s_a)|}, \quad (1)$$

$$\text{Weighted-}\beta(s_a, \mathcal{S}, \mathcal{W}_{N=3}) = \sum_{n=1, w_n \in \mathcal{W}_N}^{N} w_n \beta(s_a, \mathcal{S}, n), \quad (2)$$

$$\text{where} \sum_{n=1, w_n \in \mathcal{W}_N}^{N} w_n = 1.$$

In these equations, s_a is a given sentence, $\mathcal{S}$ is the set of currently selected anchor sentences, n-gram$(\cdot)$ generates a set of n-grams for the given text, $\mathbb{I}(\cdot)$ is an indicator function, and $w_n \in \mathcal{W}_N$ is the weight for the n-gram diversity value. We set N to 3 and define $\mathcal{W}_{N=3}$ as $\{0.6, 0.3, 0.1\}$, the weights for unigram, bigram, and trigram diversity values. Equation (1) calculates the proportion of unique n-grams in s_a compared to the n-grams in $\mathcal{S}$. Equation (2) computes the weighted average of each n-gram diversity value, computed with equation (1), for $n \in \{1, \ldots, N\}$. We select financial sentences as anchor sentences if their weighted diversity exceeds 0.6.

However, n-gram diversity alone does not guarantee semantic diversity, as it does not consider the semantic meaning of sentences. Therefore, we incorporate a cosine similarity metric based on the average FinBERT embeddings of given sentences:

$$y_{\text{FinBERT}} = \frac{z_1 \cdot z_2}{\|z_1\| \|z_2\|} \quad (3)$$

where $z_i = \text{FinBERT}(s_i)$ is the average FinBERT embedding of a sentence s_i.

We compute the similarities of s_a with sentences in $\mathcal{S}$ and take the maximum similarity score. We set different cosine similarity thresholds for 10-K filings (0.65) and earnings conference calls (0.7) to account for the varying levels of semantic richness in these documents. Financial sentences are selected as anchor sentences if their maximum similarity score is below the threshold and they meet the n-gram diversity metric condition. This similarity-based selection criterion ensures that the highest inter-similarity among selected anchor sentences is lower than the threshold, promoting semantic diversity.

A sentence is selected as an anchor if its weighted n-gram diversity exceeds 0.6 and its maximum cosine similarity score is below the respective threshold. This process continues until

150 sentences are selected from each set of 1 million sentences sampled from a specific year's 10-K filings or earnings call transcripts. If 150 sentences cannot be collected with the set thresholds, we adjust either the diversity metric threshold to 0.57 or the cosine similarity metric threshold to 0.715. As a result, a total of 3,600 anchor sentences are selected from the 24 million sentences.

2) Candidate Sentences Pairs Selection: After selecting 3,600 anchor sentences, we pair them with other sentences to form a semantically rich pool of 1.044 million sentence pairs. We employ two strategies in the pairing process, aiming to maximize semantic pattern diversity. One strategy pairs each anchor sentence with the consecutive sentences before and after it [13]. Another strategy focuses on two dimensions: temporal & source diversity, and semantic diversity. For temporal & source diversity, we pair sentences from the same or different years and sources relative to the anchor sentences. For semantic diversity, we use cosine similarities of average FinBERT embeddings to construct pairs with various similarity ranges. We consider incorporating an n-gram overlap metric based on the n-gram diversity metric but decide against it due to computational time constraints.

The second sampling strategy involves constructing two sentence pairs within the same similarity range for each anchor sentence using the 24 sets of 1 million sentences from each year's 10-K filings or earnings call transcripts. We set six similarity bins, each corresponding to one-sixth of the interval between 0 and 1. These bins match the 0-5 similarity scores in our FinSTS dataset. The cosine similarity function outputs scores between -1 and 1, but most of the resulting scores are between 0 and 1. We interpret scores below $\frac{1}{6}$ as indicating no similarity. This process results in 1.044 million sentence pairs, with 7,200 based on the first strategy and 1,036,800 $(3,600 \times 2 \times 6 \times 24)$ based on the second strategy. Overall, this approach ensures the diversity of candidate sentence pairs.

3) Construction of the FinSTS Dataset: We construct a FinSTS gold set containing 4,000 annotated sentence pairs. Initially, we randomly select sentence pairs from the pool of unlabeled candidate sentence pairs for annotation. However, an inspection of these pairs reveals an unbalanced distribution of similarity levels, with most pairs concentrated at the lower end of the 0-5 scale. To investigate the source of this imbalance, we evaluate the similarity distributions of pairs based on the cosine similarity scores from average FinBERT embeddings. We sample four sets of 100 pairs with extreme ranges of these similarity scores: less than 0.05, less than 0.1, greater than 0.9, and greater than 0.95. The analysis shows that pairs with scores above 0.95 have more diverse similarity levels, while pairs in other ranges are predominantly at the lower end of the scale, aligning with the overall distribution of the initial inspection.

To balance the dataset, we focus on the 12,076 pairs with scores above 0.95 among the candidate sentence pairs. We aim to include pairs with less lexical overlap to diversify the patterns of sentence pairs in the gold set. To determine lexical overlap levels, we compute cosine similarity scores of term frequency-inverse document frequency (TF-IDF) representations (abbreviated as TF-IDF scores). For each anchor sentence, we select the two pairs with the lowest TF-IDF scores. We then include more pairs with less lexical overlap by choosing those with TF-IDF scores less than 0.3, set after examining actual lexical overlap levels and TF-IDF scores of many pairs. In fact, pairs constructed with many anchor sentences have TF-IDF scores higher than 0.3. This selection strategy results in 2,771 pairs.

The 2,771 pairs are combined with a subset of those randomly selected 4,000 pairs to construct the final gold set. The gold set consists of 4,000 pairs, labeled with GPT-4 model and later validated by four finance experts (see section III-D). The gold set is split into a development set of 2,001 pairs and a test set of 1,999 pairs. The distribution of labels in the development and test sets is shown in Table II.

TABLE II
DISTRIBUTION OF LABELED SENTENCE PAIRS IN FINSTS

Label	# Pairs in Dev Gold Set	# Pairs in Test Gold Set
0	655	687
1	221	167
2	43	72
3	178	362
4	696	385
5	208	326
Total	**2,001**	**1,999**

C. Active Learning Algorithm

The active learning (AL) algorithm strategically selects the most informative sentence pairs for labeling and iteratively refines a FinSentenceBERT model. This section describes the construction of the initial training set and the AL algorithm framework.

1) Initial AL Training Set Construction: The AL algorithm starts with an initial training set. This initial training set should be diverse, informative, and representative of the challenges posed by the financial domain. We leverage the knowledge of 10 SBERT models that have the best performance on both general-domain STS and FinSTS (see Table IV). As noted earlier, these SBERT models are pre-trained language models (PLMs) fine-tuned on a combination of STS and Natural Language Inference (NLI) datasets, which enable them to produce high-quality sentence embeddings.

For each of the 1.04 million unlabeled candidate sentence pairs, we compute 10 cosine similarity scores using the sentence embeddings generated by these top SBERT models. We then select pairs for labeling using two criteria based on the ensemble of SBERT models: the top 4,000 variances and the top 4,000 prediction ranges of the 10 similarity scores. Variance quantifies the level of disagreement among the 10 similarity scores. A high variance indicates that the SBERT models, despite their strong performance on general-domain STS tasks, are uncertain about the predicted sentence embeddings and similarity level for a given financial pair. This uncertainty suggests that the pair contains informative signals

that can potentially improve the FinSentenceBERT model's understanding of financial language semantics. Similarly, a wide prediction range of the 10 similarity scores signifies a high level of deviation and uncertainty in the models' predictions, and the potential value of the corresponding sentence pairs for the AL process. These informative sentence pairs constitute the initial AL training set. Next, we introduce the AL algorithm that iteratively refines the model while minimizing the required annotation effort.

Algorithm 1 Active Learning for Sentence Embedding

Input: unlabeled sentence pair set $\mathcal{U} \leftarrow \{(s_1^u, s_2^u)\}_{u=1}^{|\mathcal{U}|}$, initial train STS set $\mathcal{L}_{\text{train}}^0 \leftarrow \{(s_1^i, s_2^i, y^i)\}_{i=1}^{|\mathcal{L}_{\text{train}}^0|}$, gold development STS set $\mathcal{L}_{\text{dev}}$, gold test STS set $\mathcal{L}_{\text{test}}$, 10 SBERT similarity matrix $\hat{\mathbf{Y}}_{10\text{ SBERTs}} \in \mathbb{R}^{10\times|\mathcal{U}|}$, selection strategy $\phi(\cdot)$, max number of pairs to select N, max iterations T, number of training epochs E, batch size B, weight decay λ, learning rate η, Convergence threshold ϵ.
Output: Train STS set $\mathcal{L}_{\text{train}}$, FinSentenceBERT $\mathcal{M}^*$
Initialize: a PLM (e.g., FinBERT) with mean pooling layer $\mathcal{M}^{*(0)}$, Best validation score $\rho_{\text{dev}}^{*(0)} \leftarrow 0$, Index set of unlabeled sentence pairs $\mathcal{I} \leftarrow \{1, 2, \ldots, |\mathcal{U}|\}$
for each iteration $t = 1$ to T **do**
  $K \leftarrow \lceil \frac{|\mathcal{L}_{\text{train}}^{t-1}|}{B} \rceil$ ▷ Number of training steps per epoch
  $V \leftarrow \frac{K}{10}$ ▷ Evaluation step
  $\rho_{\text{dev}}^{*(t)} \leftarrow 0$ ▷ Record the current best validation score
  $\mathcal{M}^{(t,0,0)} \leftarrow \mathcal{M}^{*(t-1)}$ ▷ Align model notations
  for each epoch $e = 1$ to E **do**
    for batch $\mathcal{B} \subset \mathcal{L}_{\text{train}}^{(t-1)}$ of size B **and** $k = 1{:}K$ **do**
      $j \leftarrow K(e-1) + k$ ▷ Training steps per iter
      $\mathcal{M}^{(t,e,k)} \leftarrow \text{Train}(\mathcal{M}^{(t,e-1,k-1)}, \mathcal{B}_k^{(t,e)}, \eta, \lambda)$
      if $j \mod V = 0$ **then** ▷ For every Vth step
        $\rho_{\text{dev}} \leftarrow \text{Evaluate}(\mathcal{M}^{(t,e,k)}, \mathcal{L}_{\text{dev}})$
      end if
      if ρ_{dev} exists **and** $\rho_{\text{dev}} > \rho_{\text{dev}}^{*(t)}$ **then**
        $\rho_{\text{dev}}^{*(t)} \leftarrow \rho_{\text{dev}}$
        $\mathcal{M}^{*(t)} \leftarrow \mathcal{M}^{(t,e,k)}$ ▷ Save the best model
      end if
    end for
  end for
  $\rho_{\text{dev}} \leftarrow \text{Evaluate}(\mathcal{M}^{(t,E,K)}, \mathcal{L}_{\text{dev}})$
  if $\rho_{\text{dev}} > \rho_{\text{dev}}^{*(t)}$ **then**
    $\rho_{\text{dev}}^{*(t)} \leftarrow \rho_{\text{dev}}$
    $\mathcal{M}^{*(t)} \leftarrow \mathcal{M}^{(t,E,K)}$
  end if
  $\rho_{\text{test}}^{*(t)} \leftarrow \text{Evaluate}(\mathcal{M}^{*(t)}, \mathcal{L}_{\text{test}})$ ▷ Evaluate on test set
  $\hat{\mathbf{y}}_{\mathcal{M}^{*(t)}} \leftarrow \text{CosineSimilarity}(\mathcal{M}^{*(t)}, \mathcal{U}) \in \mathbb{R}^{1\times|\mathcal{U}|}$ ▷ Compute cosine similarity scores of unlabeled pairs
  $\mathcal{U}_{\text{top}}, \mathcal{I}_{\text{top}} \leftarrow \text{SelectTopN}(\phi(\hat{\mathbf{Y}}_{10\text{ SBERTs}}, \hat{\mathbf{y}}_{\mathcal{M}^{*(t)}}), N)$ ▷ Select top N pairs
  $\mathcal{L}_{\text{top}} \leftarrow \text{GPTLabel}(\mathcal{U}_{\text{top}})$ ▷ Label new pairs
  $\mathcal{L}_{\text{train}}^t \leftarrow \mathcal{L}_{\text{train}}^{t-1} \cup \mathcal{L}_{\text{top}}$ ▷ Update training set
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{U}_{\text{top}}$ ▷ Update unlabeled pair pool
  $\hat{\mathbf{Y}}_{10\text{ SBERTs}} \leftarrow \hat{\mathbf{Y}}_{10\text{ SBERTs}[:,\mathcal{I}\setminus\mathcal{I}_{\text{top}}]}$ ▷ Update the matrix
  if $|\rho_{\text{dev}}^{*(t)} - \rho_{\text{dev}}^{*(t-1)}| < \epsilon$ **then break** ▷ Convergence test
  end if
end for

2) Active Learning Algorithm: The AL algorithm is outlined in Algorithm 1. The algorithm requires several inputs: a pool of unlabeled sentence pairs ($\mathcal{U}$), an initial training set ($\mathcal{L}_{\text{train}}^0$), a gold set ($\mathcal{L}_{\text{dev}}, \mathcal{L}_{\text{test}}$) for evaluation and testing, similarity scores from 10 SBERT models ($\hat{\mathbf{Y}}_{10\text{ SBERTs}}$) for each unlabeled pair, a scoring function (ϕ) to assess the informativeness of each pair, and various hyperparameters controlling the learning process.

At each iteration (t), the algorithm trains the FinSentenceBERT model ($\mathcal{M}^{(t,e,k)}$) using the current AL training set ($\mathcal{L}_{\text{train}}^{t-1}$) and evaluates its performance on the gold development set ($\mathcal{L}_{\text{dev}}$) every V steps. The best-performing model ($\mathcal{M}^{(t)}$) and its corresponding validation score ($\rho_{\text{dev}}^{(t)}$) are saved. The algorithm then uses the fine-tuned model to compute similarity scores ($\hat{\boldsymbol{y}}_{\mathcal{M}^{*(t)}}$) for the unlabeled sentence pairs and selects the top N pairs ($\mathcal{U}_{\text{top}}$) with the highest selection scores from the scoring function (ϕ) or the highest mean squared error (MSE) between the FinSentenceBERT and SBERT scores in this study. These pairs are considered the most informative and are labeled using the GPT-4 model ($\mathcal{L}_{\text{top}} \leftarrow \text{GPTLabel}(\mathcal{U}_{\text{top}})$) and added to the training set for the next iteration.

The use of the GPT-4 model for labeling in the AL algorithm offers several advantages over traditional human annotation. GPT-4 can efficiently handle domain-specific terminology and sentence complexity, and can significantly reduce annotation time and cost. Additionally, GPT-4 provides labeling explanations for verification of label accuracy. To validate the quality of GPT-4 annotations, we compare them with human annotations, as detailed in section III-D. The analysis shows that GPT-4 annotations are highly consistent with human judgments, and confirms the reliability of using GPT-4 for labeling.

The algorithm continues the iterative process until one of two stopping criteria is met: reaching the maximum number of iterations (T) or the improvement in the model's performance on the development set falling below the convergence threshold (ε). The final outputs include an enhanced training set ($\mathcal{L}_{\text{train}}$) that captures various aspects of the unlabeled sentence pair pool and a fine-tuned FinSentenceBERT model ($\mathcal{M}^*$) capable of producing semantically meaningful sentence embeddings in the financial domain.

D. Quality of GPT-4 Labeling for Financial STS Tasks

To validate the quality of GPT-4 labels, we compare GPT-4 annotations with human annotations. We randomly split the 4,000 pairs in FinSTS annotated by GPT-4 into four parts and have them annotated by four human experts. These experts are finance students who independently provide similarity scores for each pair. The analysis shows that GPT-4 annotations are highly consistent with human judgments. For 92% of the sentence pairs, the GPT-4 label falls within 1 point of the human label on the 0-5 similarity scale. The correlation

coefficient between GPT-4 and human labels is 0.91, demonstrating that GPT-4 can reliably annotate financial sentence pairs, comparable to human experts.

Additionally, using GPT-4 for labeling offers significant efficiency and cost benefits. With the GPT-4 turbo API, annotating a batch of 10 sentence pairs takes less than 30 seconds and costs approximately $0.025. In contrast, traditional human annotation methods, such as using Amazon Mechanical Turk (AMT) workers, can cost around $1 per batch of 20 pairs [1]. Moreover, GPT-4 can provide labeling explanations alongside the similarity scores, facilitating easy verification of label accuracy and traceability of the labeling logic. Obtaining such explanations from human annotators would require significant additional time and cost.

To ensure the robustness of GPT-4 annotations, we implement several measures. We adapt the annotation guidelines from the STS benchmark dataset [1] and incorporate them into the prompts, along with example sentence pairs and their expected labels. This guides GPT-4 towards providing annotations consistent with the desired similarity scale. We also employ prompt engineering techniques, such as shuffling the order of sentence pairs and providing clear instructions, to mitigate potential biases and inconsistencies in GPT-4's responses.

IV. Experiments and Results

TABLE III
PERFORMANCE OF MODELS ON FINSTS

Model	Spearman Correlation	
Mean-Pooling PLMs		
BERT	75.58 ± 1.08	
distillRoBERTa	74.44 ± 1.01	
FinBERT	**76.55 ± 0.88**	
Top General-domain SBERTs		
stsb-mpnet-base-v2	**82.68 ± 0.67**	
stsb-roberta-base	82.06 ± 0.67	
nli-mpnet-base-v2	81.98 ± 0.66	
PLMs Trained on STSb		
BERT+STSb	**80.79 ± 0.77**	
FinBERT+STSb	80.59 ± 0.68	
Fine-tuned with AL		
	Before AL	**After AL**
BERT	75.58 ± 1.08	76.59 ± 0.96
distillRoBERTa	74.44 ± 1.01	79.00 ± 0.91
FinBERT	76.55 ± 0.88	79.92 ± 0.83
FinBERT+STSb	80.59 ± 0.68	80.97 ± 0.67
SBERT (stsb-mpnet-base-v2):	**82.68 ± 0.67**	**82.89 ± 0.65**

We implement the active learning (AL) algorithm with the following hyper-parameters as specified in Algorithm 1:

1) The maximum number of selected pairs (N) is set to 200 per iteration for diversity in training set updates;
2) The total number of iterations (T) is set to 15 to balance convergence speed and stability;
3) The number of epochs (E) is set to 2 to prevent overfitting and maintain computational efficiency;
4) The batch size (B) is set to 16, the maximum allowed by our 40GB GPU memory;
5) The weight decay rate (λ) is set to 0.01 to prevent overfitting;
6) The learning rate (η) is set to 2e-05, a common choice for fine-tuning general-domain SBERT models, balancing convergence speed and stability; and
7) The model convergence rate (ϵ) is set to 1e-05.

Table III compares various sentence transformer models on the FinSTS dataset labeled by GPT-4. The 95% confidence intervals, computed using 500 bootstrapped samples from the test set, reflect model variability and ensure robust comparison.

Our initial experiments with the mean-pooled pre-trained FinBERT model as the FinSentenceBERT model in the AL algorithm show significant improvement, with the test correlation score increasing from an average of 76.55% to 79.92%. However, the model encounters a performance plateau around 80%, even after hyper-parameter tuning. We observe that nine out of ten SBERT models achieve test correlation scores higher than 80% (see Table IV), with the top three models shown in Table III for comparison.

TABLE IV
PERFORMANCE OF SBERT MODELS ON STS AND FINSTS

Model Name	STSb	FinSTS
stsb-mpnet-base-v2	88.57	82.68
stsb-roberta-base-v2	87.21	81.37
nli-mpnet-base-v2	86.53	81.97
stsb-distilroberta-base-v2	86.41	81.35
stsb-roberta-large	86.39	81.55
nli-roberta-base-v2	85.54	79.88
stsb-roberta-base	85.44	82.05
stsb-bert-large	85.29	81.9
stsb-distilbert-base	85.16	80.81
stsb-bert-base	85.14	81.21

To investigate whether this bottleneck is specific to FinBERT, we test two other BERT-family models: the pre-trained uncased BERT base model [7] and the distilled version of the case-sensitive RoBERTa base model [24]. While both models show improvements after running the AL algorithm, they yield lower test correlation scores than FinBERT.

Observing that eight out of the nine best-performing SBERT models are fine-tuned using the STS benchmark dataset, we incorporate this dataset into our training process. We train the mean-pooled FinBERT (FinBERT+STSb) and uncased BERT base (BERT+STSb) models for 10 epochs on the STS benchmark dataset's training set, evaluating them with the financial development gold set during training. Both models surpass the 80% test performance mark, with BERT+STSb slightly outperforming FinBERT+STSb. Running the AL algorithm with FinBERT+STSb further improves its test performance from an average of 80.59% to 80.97%, outperforming BERT+STSb.

Despite these improvements, the fine-tuned models still underperform compared to the top three SBERTs. To address

this, we employ the AL algorithm with the best-performing stsb-mpnet-base-v2 model, which slightly enhances its average performance from an average of 82.68% to 82.89%, achieving the best results in our experiments. We denote this best-performing model as FinSentenceBERT. Table III shows that AL further enhances model performance and makes the fine-tuned models more robust as implied by their narrower confidence intervals after running AL.

V. Conclusion

We introduce the FinSTS dataset and FinSentenceBERT model to advance semantic textual similarity in finance. FinSTS captures diverse financial texts and sets a new benchmark for evaluating STS models in finance. FinSentenceBERT, developed using an active learning algorithm with GPT-4 annotations, is validated against human judgments, confirming its effectiveness as a cost-efficient alternative to manual labeling. Experimental results show FinSentenceBERT's superiority in capturing financial semantics, outperforming other benchmark models. Notably, a general-domain SBERT model trained with the proposed active learning method surpasses FinBERT-based models. The active learning method consistently improves model performance and narrows down the 95% confidence intervals on FinSTS test set, highlighting its effectiveness.

By generating high-quality vector representations for financial texts, FinSentenceBERT supports various financial NLP tasks, including information retrieval, topic modeling, text regression, classification, and sentiment analysis [25]. In unsupervised learning, the embeddings facilitate clustering and similarity analysis, such as comparing financial disclosure similarities between firms [26]. In supervised learning, the embeddings can train models for tasks like bankruptcy prediction [27], fraud detection, and risk factor prediction. FinSentenceBERT's strong performance suggests its potential to significantly enhance financial text processing and decision-making.

Our framework for developing domain-specific language models can be adapted to other specialized domains. Future research directions include evaluating learned representations for downstream financial NLP tasks and extending the active learning approach to other data types.

References

[1] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 1–14, arXiv:1708.00055 [cs].

[2] S. H. Teoh, "The promise and challenges of new datasets for accounting research," *Accounting, Organizations and Society*, vol. 68-69, pp. 109–117, Jul. 2018.

[3] G. Hoberg and G. Phillips, "Text-based network industries and endogenous product differentiation," *Journal of political economy*, vol. 124, no. 5, pp. 1423–1465, 2016.

[4] K. Li, F. Mai, R. Shen, and X. Yan, "Measuring corporate culture using machine learning," *The Review of Financial Studies*, vol. 34, no. 7, pp. 3265–3315, 2021.

[5] M. Fathollahi, J. Harford, and S. Klasa, "Anticompetitive effects of horizontal acquisitions: The impact of within-industry product similarity," *Journal of Financial Economics*, vol. 144, no. 2, pp. 645–669, 2022.

[6] T. Loughran and B. McDonald, "Textual analysis in finance," *Annual Review of Financial Economics*, vol. 12, pp. 357–375, 2020.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[8] A. H. Huang, H. Wang, and Y. Yang, "FinBERT: A Large Language Model for Extracting Information from Financial Text*," *Contemporary Accounting Research*, vol. 40, no. 2, pp. 806–841, 2023.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Aug. 2019, arXiv:1908.10084 [cs].

[10] L. E. Dor, A. Halfon, A. Gera, E. Shnarch, L. Dankin, L. Choshen, M. Danilevsky, R. Aharonov, Y. Katz, and N. Slonim, "Active learning for bert: an empirical study," in *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, 2020, pp. 7949–7962.

[11] W. Liang, G. A. Tadesse, D. Ho, L. Fei-Fei, M. Zaharia, C. Zhang, and J. Zou, "Advances, challenges and opportunities in creating data for trustworthy ai," *Nature Machine Intelligence*, vol. 4, no. 8, pp. 669–677, 2022.

[12] Y. Wang, N. Afzal, S. Fu, L. Wang, F. Shen, M. Rastegar-Mojarad, and H. Liu, "MedSTS: a resource for clinical semantic textual similarity," *Language Resources and Evaluation*, vol. 54, no. 1, pp. 57–72, Mar. 2020.

[13] X. Guo, H. Mirzaalian, E. Sabir, A. Jaiswal, and W. Abd-Almageed, "CORD19STS: COVID-19 Semantic Textual Similarity Dataset," Nov. 2020, arXiv:2007.02461 [cs].

[14] N. Evangelopoulos, X. Zhang, and V. R. Prybutok, "Latent Semantic Analysis: five methodological recommendations," *European Journal of Information Systems*, vol. 21, no. 1, pp. 70–86, Jan. 2012.

[15] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," Oct. 2020, arXiv:2010.02559 [cs].

[16] D. Wright, J. Pei, D. Jurgens, and I. Augenstein, "Modeling information change in science communication with semantically matched paraphrases," 2022.

[17] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.

[18] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.

[19] J. Su, J. Cao, W. Liu, and Y. Ou, "Whitening sentence representations for better semantics and faster retrieval," *arXiv preprint arXiv:2103.15316*, 2021.

[20] K. Wang, N. Reimers, and I. Gurevych, "Tsdae: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 671–688.

[21] A. H. Huang, R. Lehavy, A. Y. Zang, and R. Zheng, "Analyst Information Discovery and Interpretation Roles: A Topic Modeling Approach," *Management Science*, vol. 64, no. 6, pp. 2833–2855, Jun. 2018.

[22] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[23] X. Zeng, S. Garg, R. Chatterjee, U. Nallasamy, and M. Paulik, "Empirical Evaluation of Active Learning Techniques for Neural MT," in *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, C. Cherry, G. Durrett, G. Foster, R. Haffari, S. Khadivi, N. Peng, X. Ren, and S. Swayamdipta, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 84–93.

[24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[25] N. Webersinke, "Natural Language Processing meets Accounting and Finance: Review and Performance Comparison of Textual Analysis Approaches," Rochester, NY, Jul. 2023.

[26] G. Hoberg and C. Lewis, "Do fraudulent firms produce abnormal disclosure?" *Journal of Corporate Finance*, vol. 43, pp. 58–85, Apr. 2017.

[27] F. Mai, S. Tian, C. Lee, and L. Ma, "Deep learning models for bankruptcy prediction using textual disclosures," *European Journal of Operational Research*, vol. 274, no. 2, pp. 743–758, Apr. 2019.