J. Electr. Comput. Eng. Innovations, 13(2): 403-416, 2025

Journal of Electrical and Computer Engineering Innovations (JECEI)

Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Enhancing Multi-Entity Detection and Sentiment Analysis in Financial Texts with Hierarchical Attention Networks

L. Hafezi, S. Zarifzadeh*[*], M. R. Pajoohan

Computer Engineering Department, Faculty of Engineering, Yazd University, Yazd, Iran.

Article Info

Article History:
Received 18 November 2024
Reviewed 21 January 2025
Revised 04 February 2025
Accepted 05 February 2025

Keywords:
Hierarchical attention network
Entity detection
Sentiment analysis

[*]Corresponding Author's Email Address:
szarifzadeh@yazd.ac.ir

Abstract

Background and Objectives: Detecting multiple entities within financial texts and accurately analyzing the sentiment associated with each is a challenging yet critical task. Traditional models often struggle to capture the nuanced relationships between multiple entities, especially when sentiments are context-dependent and spread across different levels of a document. Addressing these complexities requires advanced models that can not only identify multiple entities but also distinguish their individual sentiments within a broader context. This study aims to introduce and evaluate two novel methods, ENT-HAN and SNT-HAN, built upon the Hierarchical Attention Networks, specifically designed to enhance the accuracy of both entity extraction and sentiment analysis in complex financial documents.
Methods: In this study, we design ENT-HAN and SNT-HAN methods to address the tasks of multi-entity detection and sentiment analysis within financial texts. The first method focuses on entity extraction, where capture hierarchical relationships between words and sentences. By utilizing word-level attention, the model identifies the most relevant tokens for recognizing entities, while sentence-level attention helps refine the context in which these entities appear, allowing the model to detect multiple entities with precision. The second method is applied for sentiment analysis, aiming to classify sentiments into positive, negative, or neutral categories. The sentiment analysis model employs hierarchical attention to identify the most important words and sentences that convey sentiment about each entity. This approach ensures that the model not only focuses on the overall sentiment of the text but also accounts for context-specific variations in sentiment across different entities. Both methods were evaluated on FinEntity dataset, and the results demonstrate their effectiveness, with significantly improving the accuracy of both entity extraction and sentiment classification tasks.
Results: The ENT-HAN and SNT-HAN demonstrated strong performance in both entity extraction and sentiment analysis, outperforming the methods they were compared against. For entity extraction, ENT-HAN was evaluated against RNN and BERT models, showing superior accuracy in identifying multiple entities within complex texts. In sentiment analysis, SNT-HAN was compared to the best-performing method previously applied to FinEntity dataset. Despite the good performance of the existing methods, SNT-HAN demonstrated superior results, achieving a better accuracy.
Conclusion: The outcome of this research highlights the potential of the ENT-HAN and SNT-HAN for improving entity extraction and sentiment analysis accuracy in financial documents. Their ability to model attention at multiple levels allows for a more nuanced understanding of text, establishing them as a valuable resource for complex tasks in financial text analysis.

Introduction

In the realm of natural language processing (NLP), sentiment analysis has progressed from broad evaluations of textual sentiment to more precise, entity-level sentiment analysis, reflecting the growing need for detailed insights that businesses, social platforms, and researchers seek to better understand public opinion. This granular approach is particularly crucial in the financial domain, where sentiment analysis has become an indispensable tool for deciphering market trends, investor behavior, and overall economic sentiment [1]. By focusing on specific financial entities such as companies, stocks, or market indices within a text, entity-level sentiment analysis provides a more nuanced understanding than traditional document-level or sentence-level analyses, offering critical insights into how individual entities are perceived [2]. These insights are essential for predicting market movements and evaluating investor sentiment with a greater accuracy.

The challenge in this domain is twofold: first, accurately identifying financial entities within complex and often ambiguous language; second, determining the sentiment directed at each entity, which can vary significantly within different contexts. Traditional methods, which often rely on rule-based approaches or classical machine learning, struggle to capture the intricate dependencies and nuances present in financial texts, especially when multiple entities are discussed with differing sentiments. For example, in this paragraph: "Among the highlights of this week, Macy's raised its annual profit forecast, easing some investor worries over consumer spending after recent disappointments on the earnings front from Walmart and other big names.", several financial entities are referenced, but their sentiments differ, with a positive sentiment for Macy's and a negative sentiment for Walmart.

To address the above challenges, this paper proposes utilizing the Hierarchical Attention Network (HAN) [3], suggesting the ENT-HAN method for multi-entity detection and the SNT-HAN method for sentiment analysis. These methods are designed to model hierarchical structures in the text, capturing both word-level and sentence-level dependencies. The model consists of two key attention mechanisms: one at the word level and another at the sentence level. At the word level, the attention mechanism identifies the most relevant words within each sentence, assigning higher weights to words that contribute more significantly to the task at hand, such as entity extraction or sentiment detection. These weighted word representations are then aggregated into a sentence representation. At the sentence level, the attention mechanism operates similarly, focusing on the most informative sentences within the text, producing a document-level representation. This hierarchical structure allows the models to focus on the most critical parts of a text, enabling them to handle longer and more complex documents more effectively. The models are particularly suited for tasks involving multiple entities or sentiments, as they capture context at both granular and holistic levels.

Entity extraction plays a pivotal role in the sentiment analysis process, as it enables the precise identification of specific entities within text, allowing for a more targeted and accurate assessment of sentiment. By isolating entities, sentiment analysis can be more effectively tailored to evaluate opinions and emotions associated with particular individuals, organizations, or products, thus enhancing the overall reliability and relevance of the analysis [4]. When the critical step of accurately identifying multiple entities within a text is successfully completed, it becomes feasible to proceed to the sentiment analysis of each entity with greater ease and confidence. Successfully addressing the first challenge of entity extraction lays a strong foundation, allowing the second challenge—analyzing the sentiment associated with each entity—to be tackled more effectively and with higher reliability.

By applying ENT-HAN and SNT-HAN, we aim to enhance the precision of entity extraction and enhance the accuracy of sentiment analysis at the entity level. The proposed models are evaluated on FinEntity dataset[1] [5], with a comparison to existing methodologies, demonstrating its superior performance in extracting multiple entities dynamics and sentiments associated with each entity in financial texts.

Related Work

This section primarily reviews the most relevant papers on financial sentiment analysis and event-based sentiment analysis.

A. Financial Sentiment Analysis

Sentiment analysis has evolved an essential tool within the financial industry, providing valuable insights into market trends, investor behavior, and economic outlooks [6]. Unlike general sentiment analysis, which focuses on broader textual sentiment, financial sentiment analysis specifically targets the emotions and opinions expressed about financial entities, such as companies, stocks, and market indices. The unique characteristics of financial texts—often laden with technical jargon, abbreviations, and context-dependent phrases—demand specialized approaches that can accurately capture and interpret sentiment within this domain. Over the past decade, numerous studies have explored various methods for extracting and analyzing sentiment in financial contexts, from traditional machine learning techniques to more recent advancements in natural language processing and deep learning. This section examines the most influential works that have shaped the

[1] The FinEntity dataset is publicly accessible at https://github.com/yixuantt/FinEntity

current landscape of financial sentiment analysis, highlighting key methodologies and their impact on the field.

One of the pioneering studies applying deep learning techniques to financial polarity analysis was conducted by Kraus and Feuerriegel [7]. They employed a Long Short-Term Memory (LSTM) neural network to analyze ad-hoc company announcements and predict stock market movements, demonstrating that their method outperformed conventional machine learning approaches. Several other research efforts have explored diverse neural network architectures for financial sentiment analysis. Sohangir et al. [8] tested diverse neural network models on a StockTwits dataset and discovered that Convolutional Neural Networks (CNNs) yielded the highest level of performance. Lutz et al. [9] utilized doc2vec to generate sentence embeddings from company-specific announcements and employed multi-instance learning to forecast stock market results. Additionally, Maia et al. [10] combined text simplification with LSTM networks to classify sentences from financial news by sentiment, attaining cutting-edge outcomes in the Financial PhraseBank dataset.

While advanced deep learning approaches and specialized language models have been widely adopted to enhance sentiment analysis in finance, tailored versions of BERT for specific fields, such as FinBERT [11], have markedly enhanced financial sentiment analysis by being finely tuned for financial texts, leading to improved reliability and accuracy. Fatouros et al. [12] used a zero-shot prompting method, they evaluate various ChatGPT prompts on a carefully selected dataset of forex news headlines for sentiment classification. Moreover, they investigate the relationship between predicted sentiment and market returns as an additional evaluation metric. In comparison to FinBERT, a well-regarded model for financial sentiment analysis, ChatGPT demonstrated roughly 35% better performance in sentiment classification and a 36% stronger correlation with market returns. Ardekani et al. [13] developed FinSentGPT, a financial sentiment forecasting model built on a refined version of ChatGPT. Evaluating it on U.S. media news and a multilingual dataset from the European Central Bank, they found that FinSentGPT matches the performance of a top English finance sentiment model, outperforms a traditional machine learning model, and accurately predicts sentiment across different languages. This suggests that sophisticated large language models can provide adaptable and context-sensitive financial sentiment analysis across languages.

Luo and Mo in their paper [14] investigated sentiment towards the 45th President of the United States in news articles employing a novel entity sentiment analysis model known as the Negative Sentiment Smoothing Model (NSSM). The NSSM model adjusts sentiment scores by accounting for Negative Associated Entities (NAEs), that are entities linked to negative sentiments within the data. Three versions of the NSSM model (NSSM-A, NSSM-B, and NSSM-C) were developed using a smoothing algorithm. The study focused on "Trump" as the target entity and assessed the effectiveness of the NSSM models on a dataset comprising 10,993 paragraphs of news related to the target entity, gathered from CNN, FOX, and NPR over a three-month span from July 1, 2019, to September 30, 2019. The highest accuracy was achieved by NSSM-B, with an accuracy rate of 85.96%.

B. Entity-based Sentiment Analysis

Effective entity extraction is crucial in the sentiment analysis process, as it enables the precise identification of the subjects or entities to which sentiments are directed. By accurately extracting entities, sentiment analysis can yield more targeted and contextually relevant insights, thereby improving the overall accuracy and reliability of the analysis. This section will examine the most influential works in the domain of entity-based sentiment analysis.

Poria et al. [15] proposed an innovative deep learning approach for entity extraction in opinion mining, a critical task in sentiment analysis that focuses on identifying specific targets of opinions within a text, such as the attributes of a product or service being evaluated. They utilized a 7-layer deep convolutional neural network to classify each word in opinionated sentences as either an aspect or a non-aspect word. To further refine this process, they incorporated a set of linguistic patterns into the neural network. This hybrid classifier, coupled with a word-embedding model, was designed for sentiment analysis. The dataset used for training and testing spanned two domains: Laptop and Restaurant. The entity extraction framework achieved F-scores of 82% for the laptop domain and 87% for the restaurant domain.

Zhao et al. [16] introduce a method for sentiment analysis and key entity detection that utilizes BERT, tailored specifically for mining financial texts and analyzing public opinion on social media platforms. Their approach begins by using a pre-trained model to perform sentiment analysis, after which key entity detection is approached as a sentence matching or Machine Reading Comprehension (MRC) task at various levels of granularity, with a primary focus on identifying negative sentiment. Their approach employed RoBERTa as the pre-training model. Furthermore, they found that fine-tuning the pre-trained model yields better results than using it to generate sentence-level vectors for downstream models in their specific task. In the end, incorporating ensemble methods and focal loss further enhances performance to a certain degree. The experimental results show F1 scores of 95% and 85% on two publicly available financial negative entity recognition datasets, indicating that their method significantly outperforms traditional approaches. Additionally, this method demonstrates 96% accuracy in sentiment analysis.

Li et al. [17] introduce an innovative approach called the Twin Towers End-to-End model (TTEE) to address the Target and Aspect-Sentiment Detection (TASD) challenge. The TTEE model simplifies the complex TASD task by employing an end-to-end multi-task framework that concurrently handles target detection and aspect-sentiment classification. It utilizes a twin towers architecture, based on BERT or its advanced variants, to effectively separate the context from the given aspects, thereby reducing redundant calculations and significantly enhancing computational efficiency. This approach provides a distinct advantage in identifying implicit target entities and their associated aspects and sentiments within the context, without the need for additional model architectures. The highest F1 measure achieved for entity extraction with this model is 68.39%.

Wan et al. [18] propose a Span-based Multi-Modal Attention Network (SMAN) to address the joint task of entity and relation extraction. Although this study performs entity extraction on the data, the results are not utilized for sentiment analysis. Nonetheless, due to its focus on entity extraction, it has been included in this section. In the Entity Recognition stage, the model identifies two types of features: entity features (spans) represented by span units and contextual features (tokens) derived from the surrounding text. During the Relation Extraction (RE) stage, the model incorporates three types of features: entity features (spans), entity type features (labels), and contextual features (tokens). To enhance the modeling of interactions between these modalities, SMAN first generates unique representations for span units while capturing high-dimensional contextual features using a cloze mechanism. This mechanism allows the model to mask central span units, effectively learning contextual dependencies. Furthermore, entity type labels assist in refining relationship predictions, simulating the human reasoning process of associating entity types with potential relationships.

The proposed SMAN architecture includes a Modal-Enhanced Attention (MEA) module designed to model the contextual dependencies within single-modal data and facilitate fine-grained interactions among multi-modal data. By stacking MEA modules, the model captures enhanced representations of text and relationships. Extensive experiments on public datasets, including SciERC, ADE, and CoNLL04, demonstrate the superior performance of SMAN. The model delivers state-of-the-art performance, demonstrating significant improvements in F1 scores for entity detection across various datasets. On the ADE dataset, which comprises 4,272 samples extracted from medical reports and includes overlapping entities, SMAN achieves impressive F1 scores of 90.95%, highlighting its strong capability to manage complex overlapping scenarios effectively.

Our Methodology

This section begins by introducing the concepts and terminology used throughout this study, as well as the specific problem that our research addresses. The following provides a detailed explanation of the ENT-HAN and SNT-HAN models used for entity extraction and sentiment analysis.

A. Terminology and Problem Statement

This paper aims to solve two distinct tasks: entity extraction and sentiment analysis. A financial dataset consisting of 979 news articles is utilized, where all entities within each article are annotated along with their corresponding sentiments. The goal is to accurately identify the entities mentioned in the text and determine the sentiment (positive, negative, or neutral) associated with each entity.

The key to achieving accurate entity sentiment analysis lies in precisely determining the boundaries of the text segment where sentiment-related words are connected to and influence the target entity. In news articles, using the paragraph as the unit boundary for entity sentiment analysis proves to be practical, as sentiment words within a paragraph are generally associated, either explicitly or implicitly, with the target entity [14].

For both tasks, the appropriate paragraph p is first selected as the input unit. From the chosen paragraph, relevant sentences s are identified, followed by extracting the most relevant words w from those sentences. These selected units are then transformed into a three-dimensional vector representation, (p, s, w). Actually, for each entity i, the paragraph p_i is selected in which the entity appears and then choose the m surrounding sentences related to that entity, $p_i = (s_1, s_2, \ldots, s_m)$, and each sentence comprises n words, $s_j = (w_1, w_2, \ldots, w_n)$. In fact, word k from sentence j in paragraph i is positioned at the coordinates (i, j, k) within the three-dimensional input vector. This representation allows the model to effectively capture the hierarchical relationships between paragraphs, sentences, and words, ensuring that each word's context within a sentence and its broader role in the paragraph are preserved during both entity extraction and sentiment analysis.

Entity detection is a sequence labeling task. We formulate entity detection as a binary classification problem, the output Y_i is typically defined as a single value that represents one of the two possible classes. Given the word w, if w is an entity, it is labeled as 1; otherwise, it is labeled as 0.

$$Y_i = \begin{cases} 0 & if\ w_i is\ not\ an\ entity \\ 1 & if\ w_i is\ an\ entity \end{cases} \tag{1}$$

To formulate the sentiment analysis problem, which consists of three classes, i.e. positive, negative, and neutral, the output is computed according to the following equation:

$$Y_i = \begin{cases} 0 & if\ sentiment\ is\ neutral \\ 1 & if\ sentiment\ is\ positive \\ 2 & if\ sentiment\ is\ negative \end{cases} \tag{2}$$

In this formulation, the model assigns a probability score to each sentiment class based on the features extracted from the input text. The class with the highest probability is selected as the predicted sentiment. This approach ensures that the classification process accounts for the nuanced differences between the three sentiments categories, leading to a more accurate representation of the emotional tone of the text.

B. Architecture Overview

The proposed ENT-HAN and SNT-HAN are built upon Hierarchical Attention Network (HAN), an advanced deep learning model designed to capture hierarchical structures in sequential data, making it particularly effective for tasks involving long and complex texts. These methods operate through two distinct layers of attention: word-level and sentence-level. First, an embedded representation of each word is generated, typically using methods like GloVe or BERT. The word-level attention mechanism then calculates a relevance score for each word in a sentence, using a combination of bidirectional GRUs and attention weights to emphasize words that are crucial for the task, such as identifying entities or sentiment indicators. These weighted word vectors are aggregated into a sentence vector, effectively summarizing the sentence.

At the next level, the sentence-level attention mechanism processes these sentence vectors, again applying a bidirectional GRU and attention mechanism to assign importance to specific sentences within a document. This enables the model to prioritize sentences that are more informative or contextually important for the task. The final output is a document-level representation that captures both fine-grained (word-level) and broader (sentence-level) context, making ENT-HAN and SNT-HAN well-suited for tasks multi-entity extraction and sentiment analysis in domains like financial text, where contextual nuances play a critical role.

The models' architecture consists of two layers. In each layer, inputs are transformed into one-dimensional vectors using a sequence encoder. An attention mechanism is then applied to these vectors, assigning higher weights to the most informative inputs. These weighted inputs are subsequently passed to the next layer. Ultimately, a Sigmoid function is utilized to produce the final output for entity extraction problem, indicating whether a word is likely to be an entity or not, and a Softmax function is used to produce the final output for the sentiment analysis problem, indicating whether an entity is positive, neutral or negative. The architecture of ENT-HAN model is depicted in the Fig. 1, and the subsequent sections will provide a more detailed explanation of this method. The architecture of the SNT-HAN method is similar to the ENT-HAN, with the primary difference being the function used in the output layer. While the ENT-HAN method addresses a binary classification problem, the SNT-HAN deals with a multi-class classification task. This distinction requires adjustments in the final layer, which are discussed in greater details in the following section.

ENT-HAN and SNT-HAN models comprise multiple sections: a word sequence encoder, a word-level attention layer, a sentence encoder, a sentence-level attention layer. The following sub-sections explain each of these parts.

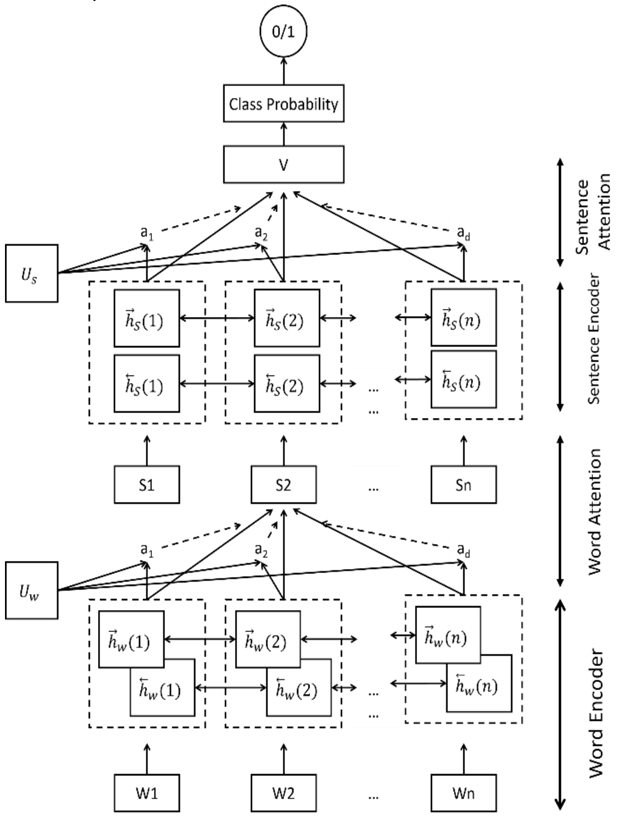

Fig. 1: ENT-HAN architecture.

The flowchart presented in Fig. 2 illustrates the step-by-step process of the ENT-HAN model for the entity detection. Each component in the flowchart represents a crucial stage in the model's operation. The process applied for sentiment analysis follows a similar approach, adhering to the same structure. This ensures consistency in how sentiments are extracted and classified across different entities and contexts within the text.

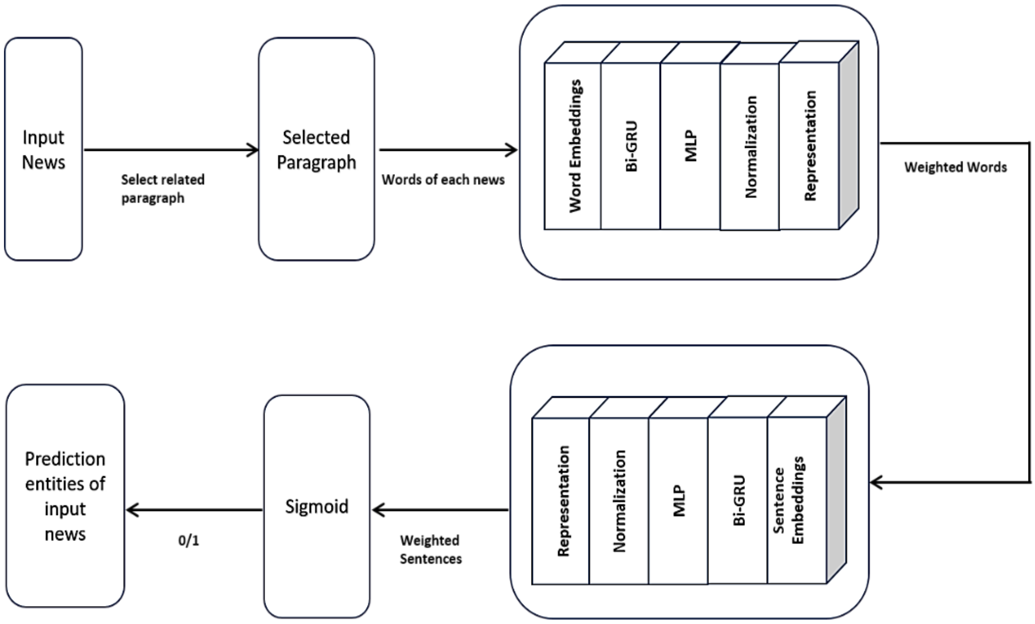

Fig. 2: Flowchart of ENT-HAN model.

Further, Algorithm 1 and Algorithm 2 outline the detailed steps of the ENT-HAN and SNT-HAN model, respectively, summarizing their core operations and logic.

Algorithm 1: Entity extraction process with ENT-HAN method

Input: a set of news paragraphs (P) belonging to all news documents, where each paragraph p_i contains sentences, and each sentence s_j contains words.

#Step 1: Preprocess inputs
$data = []$
$for\ each\ paragraph\ p_i\ in\ P$
    $for\ each\ sentence\ s_j\ in\ p_i$
        $for\ each\ word\ w_k\ in\ s_j$
            #Convert $w_k$into unique numerical representation
            $data[i, j, k] = tokenizer(w_k)$
#Step2: Map each word w_k to word embedding using a pre-trained embedding model
$embed_{seq} = GloVe(w_k)$
#Step 3: Word Encoding
$encoder_{wrd} = bidirectional\left(GRU\left(embed_{seq}\right)\right)$
$attention_{wrd} = attention\_layer(encoder_{wrd})$
#Step 4: Sentence Encoding
$encoder_{snt} = bidirectional\left(GRU(attention_{wrd})\right)$
$attention_{snt} = attention\_layer(encoder_{snt})$
#Step 5: Classification
$vector_{doc} = dence\_laye(attention_{snt})$
$entity = sigmoid(vector_{doc})$
$return\ \ entity\ \ (0\ or 1)\ \#return\ 0\ if\ the\ token\ is\ not\ an\ entity, otherwise\ return\ 1$

Output: predicted entities

Algorithm 2: Sentiment Analysis process with SNT-HAN method

Input: a set of paragraphs (P) with specified entities

#Step 1: Preprocess inputs
$data = []$
$for\ each\ entity$ in paragraph p_i
  $select\ sentences\ \{1..15\}\ and\ words\ \{1..120\}$
$for\ selected\ sentence\ s_j\ in\ paragraph\ p_i$
  $for\ selected\ word\ w_k\ in\ sentence\ s_j$
    #Convert $w_k$into unique numerical representation
    $data[i, j, k] = tokenizer(w_k)$
#Step 2: Map each word w_k to word embedding using a pre-trained embedding model
$embed_{seq} = Glove\ or\ BERT(w_k)$
#Step 3: Word Encoding
$encoder_{wrd} = bidirectional\left(GRU\left(embed_{seq}\right)\right)$
$attention_{wrd} = attention\_layer(encoder_{wrd})$
#Step 4: Sentence Encoding
$encoder_{snt} = bidirectional(GRU(attention_{wrd}))$
$attention_{snt} = attention\_layer(encoder_{snt})$
#Step 5: Classification
$vector_{doc} = dence\_laye(attention_{snt})$
$sentiment = softmax(vactor_{doc})$
$return\ 0\ or\ 1\ or\ 2\ \ \#Return\ 0\ for\ neutral, 1\ for\ positive, and\ 2\ for\ negative\ entities$

Output: sentiment (positive, negative, neutral) of each entity

1. Word Embedding

Word embedding is a technique in natural language processing where words or phrases are represented as vectors in a continuous vector space. This enables the model to position semantically similar or related words closer together, based on patterns learned from the training data [19]. GloVe and BERT represent two different approaches to word embeddings, each with distinct advantages. GloVe (Global Vectors for Word Representation) is a static embedding model, which means that each word is assigned a single vector based on co-occurrence statistics across a large corpus. This approach captures general word meanings well, but it cannot handle context-specific variations in meaning. On the other hand, BERT (Bidirectional Encoder Representations from Transformers) generates dynamic embeddings, where a word's representation depends on the surrounding context in the sentence. BERT, through its deep bidirectional architecture, captures nuanced, context-dependent meanings of words, making it more suitable for tasks like entity extraction and sentiment analysis in complex language structures. However, BERT is computationally more intensive compared to GloVe, which is faster but less capable of understanding context.

Hence, pre-trained embedded vectors are used, using GloVe in one instance and BERT in another, to provide the model with an additional advantage in terms of performance [20].

2. Word Sequence Encoder

Assume that a selected paragraph contains m sentences, and n indicates the number of words in each sentence. The word in the i^{th} sentence is denoted as w_{ij} where j$\epsilon[1, n]$. Initially, the words are transformed into vectors through an embedding matrix W_e, resulting in $x_{ij} = W_e w_{ij}$. Next, word vectors are obtained from both directions using a bidirectional GRU, which processes input sequences in both backward and forward directions. In the forward direction, the input sequence is processed from the first word to the last word, with each word's representation being influenced by the preceding words in the sequence. Simultaneously, in the backward direction, the input sequence is processed from the last word to the first word, where each word's representation is influenced by the subsequent words in the sequence. This bidirectional approach captures contextual information from both preceding and subsequent words for each word in the sequence.

The Gated Recurrent Unit (GRU) is a gating mechanism introduced in 2014 by Cho et al. [21], and further developed by Chung et al. [22]. It offers the advantage of faster computation compared to many other recurrent neural network models, which are especially effective for handling sequential data. The GRU uses two gates—the reset gate and the update gate—that control the flow of information within each unit. The update gate, denoted as z_t^j, intuitively allows the model to regulate how much of the past information from the previous state should be retained and transmitted to the new state. This gate is computed by taking a linear combination of the previous hidden state and the current input, which is then processed through a Sigmoid function:

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \tag{3}$$

where x_t is the sequence vector at time t. The reset gate, denoted as r_t^j, determines how much of the previous hidden state should be forgotten or reset. This is achieved by taking a linear combination of the previous hidden state h_{t-1} and the current input x_t , and then passing the result through an activation function. The reset gate is computed as:

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j \tag{4}$$

here, W_r represents the weight matrix, and σ is the Sigmoid activation function that ensures the gate's output remains between 0 and 1, effectively controlling the degree of resetting the previous hidden state.

The new state h_t^j at time t is a linear interpolation between the current new state $\hat{\mathrm{h}}_t^j$ and the previous state h_{t-1}^j:

$$h_t^j = \left(1 - z_t^j\right) h_{t-1}^j + z_t^j \hat{\mathrm{h}}_t^j \tag{5}$$

The candidate state $\hat{\mathrm{h}}_t^j$ is computed as follows:

$$\hat{\mathrm{h}}_t^j = \tanh(W x_t + U(r_t \odot h_{t-1}))^j \tag{6}$$

where $\odot$ is an element-wise multiplication. If r_t is zero, the model disregards the previous state.

The bidirectional nature of the above approach allows the model to gain a deeper understanding of the context in which a word appears, thereby enhancing its ability to grasp semantic relationships and meanings. This dual perspective provides a richer representation of the word, improving the model's capacity to acquire the full context of the text. Thus, we have:

$$\begin{aligned} x_{ij} &= W_e w_{ij}, j\epsilon[1,n], i\epsilon[1,m] \\ \vec{h}_{ij} &= \overrightarrow{GRU}\left(x_{ij}\right), j\epsilon[1,n], i\epsilon[1,m] \\ \overleftarrow{h}_{ij} &= \overleftarrow{GRU}\left(x_{ij}\right), j\epsilon[n,1], i\epsilon[m,1] \end{aligned} \tag{7}$$

here, x_{ij} represents the transformation of the input w_{ij} using a weight matrix W_e and $\vec{h}_{ij}$ and $\overleftarrow{h}_{ij}$ indicate the forward and backward GRU operation on x_{ij}, respectively.

3. Word Attention

In the Hierarchical Attention Network model, certain words are more critical to a sentence's meaning. To effectively combine the representations of these informative words into a sentence vector, the ENT-HAN and SNT-HAN models employ an attention mechanism. Here's a detailed breakdown of the process:

a. Bidirectional GRU (Bi-GRU):

The sentence is initially expressed as a sequence of word vectors. This sequence is processed through a bidirectional GRU, which handles the words in both their original order and reverse order. This bidirectional processing captures contextual information from both future and past words, resulting in a sequence of hidden states where each hidden state encapsulates the contextual information of a word within the sentence.

b. One-Layer MLP:

The hidden states from the Bi-GRU are fed into a one-layer Multilayer Perceptron (MLP). This layer performs a linear transformation followed by a non-linear activation function to each hidden state to calculate the importance scores for each word. Specifically, the importance score u_{ij} is computed as:

$$u_{ij} = \tanh\left(W_w h_{ij} + b_w\right) \tag{8}$$

where W_w is the weight matrix and b_w is the bias term.

c. Importance Measurement:

The output of the MLP represents the importance scores for each word in the sentence. These scores indicate the relative importance of each word within the context of the entire sentence.

d. Normalization (Softmax):

To convert the importance scores into a normalized probability distribution, the scores are passed through a Softmax function. This function ensures that the importance weights sum to 1, with higher scores translating to higher weights:

$$\alpha_{ij} = \frac{\exp(u_{ij}^T u_w)}{\sum_p \exp\left(u_{ip}^T u_w\right)} \tag{9}$$

here, u_w is a vector representing the weights, and the denominator normalizes the weights across all words in the sentence.

e. Sentence Representation:

The normalized importance weights α_{ij} are used to compute the sentence representation by taking a weighted sum of the word vectors. The sentence vector s_i is given by:

$$s_i = \sum_p \alpha_{ip} h_{ip} \tag{10}$$

where h_{ip} represents the word vectors, and α_{ip} are the normalized importance weights. This aggregation focuses on the most relevant words, resulting in a sentence vector that effectively captures the key information.

This mechanism enables the model to concentrate on the most significant words when generating a representation for the entire sentence.

4. Sentence Sequence Encoder

The same procedure used for encoding words is applied to the derived sentence vectors to create the document vector. A bidirectional GRU is employed to encode the sentences:

$$\begin{aligned} \overrightarrow{h}_i &= \overrightarrow{GRU}(s_i), i\epsilon[1, m] \\ \overleftarrow{h}_i &= \overleftarrow{GRU}(s_i), i\epsilon[m, 1] \end{aligned} \tag{11}$$

For getting an annotation of sentence i, $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ must be concatenated, i.e., $h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$. h_i encapsulates the neighboring sentences surrounding sentence i while still concentrating on this sentence.

In this part of ENT-HAN and SNT-HAN model, GRU-based sequence encoder is the same as the one applied in the word encoder.

5. Sentence Attention

Each sentence in a news article conveys a distinct semantic meaning, hence it is essential to calculate attention weights for different sentences individually to emphasize those that are more critical for event detection. To compute the document vector v_k, which summarizes all the information from the sentences in a paragraph of a news article, the following formulas are used:

$$\begin{aligned} u_i &= \tanh(W_s h_i + b_s) \\ \alpha_i &= \frac{\exp(u_i^T u_s)}{\sum_p \exp(u_p^T u_s)} \\ v &= \sum_p \alpha_p h_p \end{aligned} \tag{12}$$

A transformation is applied to the hidden state h_i using learnable parameters W_s and b_s, producing an intermediate representation u_i. Then, attention weights α_i are computed by measuring the similarity between u_i and a context vector u_s, normalized over all inputs. By following this process, the document vector v effectively captures the key information from the paragraph, highlighting the sentences that contribute most to the event detection and sentiment analysis tasks.

6: Prediction in Entity Extraction

To predict the probability of binary classification (with only two classes, entity and non-entity), the Sigmoid function is applied. The Sigmoid function transforms the model's output into a probability value between 0 and 1.

$$\rho = Sigmoid(W_p d + b_p) \tag{13}$$

Specifically, Sigmoid function takes any real-valued input and transforms it into a value in range 0 and 1, allowing the model to interpret the output as the probability of a given class. If the output of the Sigmoid function is closer to 1, the model predicts the positive class (e.g., entity present); if it's closer to 0, it predicts the negative class (e.g., entity absent). This makes the Sigmoid function ideal for binary classification problems, as it converts raw predictions into easily interpretable probabilities. To optimize the model during training, the cross-entropy loss function is employed. Cross-entropy quantifies the disparity between the predicted probabilities $q(x)$ and the actual labels $p(x)$. It assesses how effectively the predicted probability distribution corresponds with the actual distribution of the labels. The cross-entropy loss function CE is defined as:

$$CE = -\sum_x p(x) \log_2 q(x) \tag{14}$$

7. Prediction in Sentiment Analysis

For sentiment analysis, however, the Softmax function is employed, as it is designed for multi-class classification with three sentiment categories: positive, negative, and neutral. The Softmax function assigns a probability to each class, ensuring that the sum of probabilities across the three sentiment classes equals 1, facilitating a more accurate sentiment prediction.

$$\rho = Softmax(W_p d + b_p) \tag{15}$$

To optimize the SNT-HAN model during training, similar to ENT-HAN, the cross-entropy loss function is employed.

Experiments

In this study, the FinEntity dataset is used, a comprehensive collection of financial texts annotated with sentiment labels. The dataset consists of 979 example paragraphs, featuring a total of 2,131 entities classified into three sentiment categories: Positive, Negative, and Neutral. Notably, approximately 60% of the paragraphs in the dataset contain multiple entities, making it particularly challenging and relevant for tasks involving complex entity extraction in financial contexts. It is possible that in a sentence containing multiple entities, the sentiment associated with each entity may differ, presenting a complex challenge that required careful attention to address. This variability in sentiment adds an additional layer of difficulty to the analysis, as it requires distinguishing between the emotional tones linked to each individual entity within the same textual context.

To extract entities from this dataset and to identify the sentiments associated with each of the entities, we employed ENT-HAN and SNT-HAN methods, respectively and we found them highly effective for performing entity extraction and sentiment analysis tasks. The ENT-HAN method demonstrated strong performance in accurately identifying and classifying entities within the financial texts of the FinEntity dataset. Its ability to handle the intricacies of financial language, including the detection of multiple entities within the same paragraph, proved to be particularly beneficial. Additionally, SNT-HAN method shows to be highly effective in analyzing the sentiments associated with these entities, delivering strong performance in accurately capturing and classifying their emotional context.

A. Dataset

This description provides an overview of the FinEntity dataset, which is a collection of paragraphs focused on financial text. Here's a summary of the key details:

- Total Examples: 979 paragraphs.
- Total Entities: 2,131 entities classified into three sentiment categories:
  - Positive Entities: 503 entities (approximately 24% of the total).
  - Negative Entities: 498 entities (approximately 23% of the total).
  - Neutral Entities: 1,130 entities (approximately 53% of the total).
- Sentiment Label Distribution: The distribution across Positive, Negative, and Neutral entities is fairly balanced, though Neutral entities form the majority.
- Entity Presence in Text: About 60% of the financial text contains multiple entities, indicating that the dataset often features paragraphs with more than one entity labeled with a sentiment.

B. Settings

Testing different values for model parameters and selecting the optimal ones is a crucial aspect of model development and fine-tuning. In this study, we systematically explored various parameter settings to identify the configurations that yield the best performance for our model.

This process involved adjusting and evaluating multiple hyper-parameters to optimize the model's accuracy and effectiveness. The selected hyper-parameters that were found to produce the most favorable results are detailed in Table 1.

Table 1: The setting of hyper-parameters

Hyper-parameter	ENT-HAN	SNT-HAN
Batch Size	56	56
Max Sentence Length	100	120
Max Sentence Number in a Paragraph	7	15
Embedding Dimensions	100	100
Validation Split	20%	20%
Epochs	10	10
GRU Dimensions	50	50
Word Dimensions	100	100
Sentence Dimensions	100	100

The evaluation metric for assessing ENT-HAN model performance is the F1 score, which provides a balanced measure of recall and precision, ensuring that our model performs well across both aspects. The evaluation metric for assessing SNT-HAN model is accuracy.

C. Experimental Results of Entity Detection

For comparison purposes, a comparative analysis is conducted between ENT-HAN method, the BERT model—a widely recognized benchmark in natural language processing tasks—and the Recurrent Neural Network (RNN) model. By evaluating our approach against these well-established models, we aim to assess the relative strengths and weaknesses of ENT-HAN method in terms of entity extraction within financial texts. This comparison provides an understanding of how our method performs in relation to both the cutting-edge BERT model and the traditional RNN approach.

BERT (Bidirectional Encoder Representations from Transformers) [23] is a transformative method in natural language processing that has significantly advanced the state of the art in various tasks, including entity extraction. Unlike traditional models that process text sequentially, BERT employs a bidirectional approach, allowing it to capture the context of a word based on both its preceding and following words. This deep contextual understanding enables BERT to more accurately identify and classify entities within a text. Its pre-training on vast amounts of text data and subsequent fine-tuning for specific tasks have made BERT particularly effective in extracting nuanced and context-sensitive entities, which is crucial for applications in complex domains like finance literature.

Recurrent Neural Networks (RNNs) [24] are a type of neural network specifically designed to handle sequential data by retaining a memory of prior inputs. This capability enables RNNs to capture temporal dependencies, making them well-suited for tasks such as language modeling, time series forecasting, and sequence classification.

The Decoding Enhanced BERT with Disentangled Attention (DeBERTa) [25], introduced by Microsoft, represents an advanced variation of the BERT architecture that has achieved significant benchmarks across numerous natural language processing tasks. This model excels particularly in tasks such as entity extraction and sentiment classification, thanks to its ability to capture intricate linguistic structures and contextual relationships within text. Unlike its predecessors, DeBERTa employs a disentangled attention mechanism, which uses two separate vectors for each token—one to represent its semantic content and the other to encode its positional information. This disentangled representation allows the model to better understand the nuanced interactions between words in various contexts.

Furthermore, the model enhances its pre-training process through an improved mask decoder, optimizing its performance on masked language modeling tasks by providing more accurate predictions for masked tokens. Collectively, these innovations enable DeBERTa to deliver state-of-the-art performance in understanding and processing complex textual data. The primary metrics utilized for comparison in our study on the entity extraction problem include F1 score, precision, recall, and accuracy [26]. These metrics are widely recognized in classification tasks, with the F1 score providing a balanced measure of recall and precision, precision evaluating the proportion of correctly identified positive instances, recall measuring the proportion of actual positive instances correctly identified, and accuracy reflecting the overall correctness of predictions. Precision refers to the ratio of true positive predictions among the total number of positive predictions made by the model. It answers the question: "Out of all the entities the model identified as relevant, how many were actually relevant?"

On the other hand, recall is the ratio of true positive predictions to the total number of actual positive instances in the dataset. It addresses the question: "Out of all the relevant entities in the dataset, how many did the model successfully identify?". The F1 score is then defined as:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + recall} \tag{16}$$

This metric is particularly valuable in situations where there is a discrepancy in the number of relevant and irrelevant cases, as it provides a more thorough assessment of model performance by taking into account both false positives and false negatives. Focusing on the F1 score ensures that our comparison captures the overall effectiveness of the models in accurately identifying and classifying entities. Accuracy measures the proportion of correctly classified instances (both positive and negative) to the total number of instances. It reflects the overall effectiveness of a model in making correct predictions. The formula is:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{17}$$

The comparative analysis evaluated the performance of ENT-HAN method against the RNN, BERT and DeBERTa model, specifically, the study focuses on the F1 score (f1) as the key metric while also incorporating precision (Prec.), recall (Rec.), and accuracy to provide a comprehensive evaluation of model performance. The results in Table 2 show that ENT-HAN model surpasses RNN, BERT, and DeBERTa by achieving higher scores across all key metrics. This demonstrates the model's ability to balance precision and recall effectively in entity extraction tasks, accurately identifying relevant entities while minimizing false positives and false negatives. The ENT-HAN model's superior F1 score highlights its enhanced capability in capturing the context-sensitive nature of financial entities compared to BERT and DeBERTa. Leveraging its advanced attention mechanism, the model prioritizes the most critical sentences and words within the content, significantly improving entity recognition performance.

Table 2: Comparison of our model with baseline model

Method	Prec.	Rec.	F1	Accuracy
RNN	73.9	80.6	77.1	60.24
BERT	85.9	92.8	89.2	75.44
DEBERTA	90.8	96.5	93.6	93.35
ENT-HAN	89.2	96.7	92.8	93.90

The performance of DeBERTa and ENT-HAN model, which exhibit similar behavior in entity extraction, was compared in terms of computational time complexity and memory usage during execution. The results of this comparison, calculated for each epoch, are presented in the Table 3. The findings indicate that while DeBERTa demonstrates high predictive accuracy, it comes with significantly higher time and space complexity compared to ENT-HAN. This trade-off highlights the potential limitations of DeBERTa for scenarios requiring efficient computation and resource utilization.

Table 3: Comparison of DeBERTa and ENT-HAN complexity

Method	Time Complexity (Sec)	Space Complexity (MB)
DeBERTa	2925.9576	9050.84
ENT-HAN	1304.4537	4548.47

This part aims to evaluate our approach against those introduced in the literature, despite the fact that none of these approaches used the same dataset, and the datasets and domains they addressed differ significantly. While this makes a precise comparison challenging, it may still provide a general perspective on the effectiveness of our method.

Table 4: Comparison of different methods evaluated on different datasets

Method	Dataset	F1 Score
CNN+Linguistic Patterns [15]	LAPTOP REVIEWS	82.32%
	Restaurant Reviews	87.17%
Ensemble-Focal-Loss [16]	2019 CCF BDCI	95.25%
	2019 CCKS	85.05%
TTEE [17]	RES15	58.94%
	Res16	68.39%
	ACOS_LAPTOP	43.94%
SMAN [18]	ADE	90.95%
ENT-HAN	FinEntity	92.80%

Importantly, as shown in Table 4, our approach demonstrates a high level of accuracy in correctly identifying multiple entities within a single sentence—a capability that is not highlighted in any of the other methods. This advantages of accurately extracting multiple entities further underscores the robustness of our method, even if direct comparisons are limited by differences in datasets and application domains.

D. Experimental Results of Sentiment Analysis

In the case of the sentiment analysis problem, we compare our findings with those presented in the original paper [5] that introduced the dataset. In that study, six different methods were applied to the dataset, which is briefly described below. The methods include BERT, BERT-CRF, FinBERT, FinBERT-CRF, ChatGPT (zero-shot), and ChatGPT (few-shot). The BERT method, as previously explained, serves as the baseline for comparison. FinBERT is a specialized variant of the BERT model, designed by Yang et al. in 2020 [27] to address the unique challenges of natural language processing within the financial domain. Unlike standard BERT, which is trained on general language corpora, FinBERT is pre-trained on large datasets of financial texts, such as earnings reports, news articles, and financial statements, enabling it to better understand domain-specific language, terminology, and context. For each token's hidden output, FinBERT applies a linear layer to perform tasks like sentiment analysis. This model is particularly effective in tasks that require a deep understanding of financial jargon and context, and when combined with Conditional Random Field (CRF) [28] layers (as in FinBERT-CRF), it further improves performance in sequence-based tasks such as sentiment classification. Few-shot [29] and zero-shot learning [30] of ChatGPT are used to perform tasks with minimal or no task-specific training data. In zero-shot learning, the model is given a task without any prior examples or training for that specific task. It relies entirely on its general knowledge and pre-training to generate a response. For instance, in sentiment analysis, a zero-shot approach would attempt to identify the sentiment of entities based solely on its understanding of language, without seeing any labeled examples beforehand. In few-shot learning, the model is provided with a small number of examples (few-shot examples) to learn the task before making predictions. This technique allows the model to better understand the structure or rules of the task with minimal data, improving its performance compared to zero-shot. In both approaches, ChatGPT leverages its extensive pre-training, but few-shot learning typically yields more accurate and reliable results, especially in complex tasks like sentiment analysis.

This study employs micro-avg, macro-avg, and weighted-avg methods for prediction, three commonly used averaging methods for precision, recall, and F1-score provide different perspectives on performance. Micro-Averaging method aggregates true positives, false positives, and false negatives across all classes and computes the metrics globally. It is effective for datasets with class imbalance, as it gives equal weight to each instance. Macro-Averaging approach calculates the metric for each class independently and then averages them. It treats all classes equally, regardless of their size, which can highlight performance disparities between minority and majority classes. Weighted Averaging technique calculates metrics for each class and averages them, weighted by the number of instances in each class. It balances the influence of each class based on its prevalence, making it useful for understanding the model's overall performance in imbalanced datasets.

Among the models applied in the aforementioned paper, the FinBERT-CRF model achieved the best performance, with a macro average of 85%. The ChatGPT method does not perform particularly well in sentiment analysis tasks, it has the lowest level of accuracy. Despite its strong language modeling capabilities, its performance in accurately classifying sentiments, especially in domain-specific texts like financial documents, tends to lag behind more specialized models. This is likely due to the lack of fine-tuning for the specific nuances and subtleties present in sentiment expressions, which can lead to lower accuracy in identifying the correct sentiment class. As a result, while ChatGPT can provide reasonable outputs, it often struggles to match the precision and reliability of models specifically trained for sentiment analysis, such as FinBERT or other fine-tuned approaches. However, our proposed SNT-HAN method demonstrates superior accuracy in sentiment analysis, outperforming the existing models. The detailed results of the comparison are provided in the Table 5.

Table 5: Comparison of baseline method and our method

Method	MicroAvg	Macro Avg	Weighted Avg
BERT	80	80	80
BERT-CRF	81	81	81
ChatGPT (zero-shot)	59	56	59
ChatGPT (few-shot)	67	65	67
FinBERT	83	83	83
FinBERT-CRF	84	85	84
SNT-HAN with GloVe Embedding	87	86	84
SNT-HAN with BERT Embedding	89	87	85

Our results highlight the effectiveness of the SNT-HAN, which enables more precise sentiment analysis. By capturing both word and sentence-level context, the SNT-HAN model has proven to be more adept at handling complex financial text, particularly when multiple entities are present within a single sentence. Overall, these findings suggest that SNT-HAN is a highly effective approach for multi-entity sentiment analysis, outperforming traditional methods in this domain. It is also evident that using BERT for word embedding, rather than GloVe, leads to an improvement in accuracy, highlighting the impact of context-aware embeddings on model performance. The results confirm that integrating attention mechanisms significantly enhances sentiment detection accuracy in financial texts.

Conclusion

In conclusion, entity extraction is a critical component of natural language processing, particularly in domains like finance where precise identification and classification of entities are essential for accurate sentiment analysis and decision-making. ENT-HAN method has proven to be highly effective in this regard. By leveraging its advanced capabilities, ENT-HAN successfully captures the complex, context-dependent relationships between entities in text, leading to superior performance as evidenced by its higher F1 score compared to other models like RNN and BERT. The results highlight the robustness of deep learning methods in addressing the challenges of entity extraction, making ENT-HAN method a valuable tool for tasks that demand high accuracy and reliability.

The SNT-HAN proves to be a highly effective approach for sentiment analysis, particularly in scenarios involving complex texts with multiple entities. One of the key strengths of SNT-HAN lies in its ability to model attention at both word and sentence levels, allowing the model to focus on the most relevant sections of text when determining sentiment. This hierarchical structure enhances the model's capacity to capture context-dependent sentiments, leading to more accurate sentiment classification, even when multiple entities are present within a single document. Furthermore, SNT-HAN’s ability to account for the varying importance of words and sentences makes it well-suited for tasks requiring nuanced understanding, such as financial sentiment analysis. Overall, SNT-HAN’s superior performance compared to traditional models highlights its robustness and adaptability in handling intricate sentiment analysis tasks.

Author Contributions

L. Hafezi and S. Zarifzadeh conceptualized and designed the study and conducted data analysis. L. Hafezi contributed to data collection, conducted statistical analyses, and wrote the initial draft of the manuscript. S. Zarifzadeh and M. Pajoohan supervised the project and provided critical feedback during manuscript development. All authors reviewed and approved the final manuscript.

Acknowledgment

We sincerely appreciate the insightful comments and constructive suggestions provided by the anonymous reviewers, which have greatly contributed to enhancing the quality and clarity of this manuscript. We are also grateful to the editor for their careful oversight throughout the review process. Furthermore, we acknowledge the editorial team of *JECEI* for their dedication and professionalism in managing the submission and publication process.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

The following abbreviations are used throughout this document for clarity and conciseness.

HAN	Hierarchical Attention Network
LSTM	Long Short-Term Memory
NSSM	Negative Sentiment Smoothing Model
TASD	Target and Aspect Sentiment Detection
TTEE	Twin Towers End-to-End
MEA	Model Enhanced Attention
GloVe	Global Vectors for Word Representation
BERT	Bidirectional Encoder Representation from Transformers
GRU	Gated Recurrent Unit
Bi-GRU	Bidirectional GRU
M LP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
DeBERTa	Decoding Enhanced BERT with disentangled Attention
CRF	Conditional Random Field

References

[1] K. Du, F. Xing, e “Financial sentiment analysis: Techniques and applications,” ACM Comput. Surv., 56(9): 1-42, 2024.

[2] J. Yu, J. Jiang, R. Xia, "Entity-sensitive attention and fusion network for entity-level multimodal sentiment classification," IEEE/ACM Trans. Audio Speech Lan. Process., 28: 429-439, 2020.

[3] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, “Hierarchical attention networks for document classification,” in Proc. the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: 1480-1489, 2016.

[4] L. Deng et al., “Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models,” in Proc. 2015 Conf. Empirical Methods in Natural Language Processing: 179-189, 2015.

[5] Y. Tang et al., “FinEntity: Entity-level sentiment classification for financial texts,” in Proc. 2023 Conference on Empirical Methods in Natural Language Processing: 15465-15471, 2023.

[6] C. Liu et al., "Large language models and sentiment analysis in financial markets: A review, datasets, and case study," IEEE Access, 12: 134041-134061, 2024.

[7] M. Kraus, S. Feuerriegel, "Decision support from financial disclosures with deep neural networks and transfer learning," Decis. Support Syst., 104: 38-48, 2017.

[8] S. Sohangir et al., "Big data: Deep learning for financial sentiment analysis," J. Big Data, 5(1), 2018.

[9] B. Lutz et al., "Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning," arXiv preprint arXiv:1901.00400, 2018.

[10] M. Maia, et al., "FinSSLx: A sentiment analysis model for the financial domain using text simplification," in Proc. 12th International Conf. Semantic Computing (ICSC): 318-319, 2018.

[11] Z. Liu et al., "FinBERT: A pre-trained financial language representation model for financial text mining," in Proc. International Joint Conf. Artificial Intelligence: 4513-4519, 2021.

[12] G. Fatouros et al., "Transforming sentiment analysis in the financial domain with ChatGPT", Mach. Learn. Appl., 14, 100508, 2023.

[13] A. M. Ardekani et al. "FinSentGPT: A universal financial sentiment engine," int. Rew. Financ. Anal., 94, 103291, 2024.

[14] M. Luo, X. Mu, "Entity sentiment analysis in the news: A case study based on Negative Sentiment Smoothing Model (NSSM)," Int. J. Inf. Manage. Data Insights, 2(1), 2022.

[15] S. Poria, E. Cambria, A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," Knowledge-Based Syst., 108: 42-49, 2016.

[16] L. Zhao, L. Li, X. Zheng, J. Zhang, "A BERT based sentiment analysis and key entity detection approach for online financial texts," in Proc. CSCWD 2021: 1233-1238, 2021.

[17] Z. Li, Y. Song, X. Lu, M. Liu, "Twin towers end to end model for aspect-based sentiment analysis," Expert Syst. Appl., 249, 2024.

[18] Q. Wan et al., "A span-based multi-modal attention network for joint entity-relation extraction," Knowledge-Based Syst., 262, 2023.

[19] A. Thomas et al., "Investigating the impact of pre-trained word embeddings on memorization in neural networks," in Proc. Text, Speech, and Dialogue, Lecture Notes in Computer Science, 2020.

[20] C. Wang et al., "A comparative study on word embeddings in deep learning for text classification," in Proc. 4th Int. Conf. Natural Language Processing and Information Retrieva: 37-46, 2021.

[21] K. Cho et al.,"On the properties of neural machine translation: encoder-decoder approaches," in Proc. English Workshop on Syntax, Semantics and Structures in Statistical Translation, 2014.

[22] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," Workshop on Deep Learning, 2014.

[23] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1: 4171-4186, 2019.

[24] D. E. Rumelhart, J. L. McClelland, "Learning internal representations by error propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations, MIT Press, 318-362, 1987.

[25] P. He, X. Liu, J. Gao, W. Chen, "DeBERTa: Decoding enhanced BERT with Disentangled Attention," in Proc. ICLR, 2021.

[26] H. Dalianis, "Evaluation metrics and evaluation," Clinical Text Mining, Springer, Cham, 2018.

[27] Y. Yang, M. C. Uy, A. Huang, "FinBert: A pretrained language model for financial communications," arXiv:2006.08097, 2020.

[28] C. Sutton, A. McCallum, "An introduction to conditional random fields," Found. Trends Mach. Learn., 4(4): 267-373, 2011.

[29] U. Lee et al., "Few-shot is enough: exploring ChatGPT prompt engineering method for automatic question generation in English education," Educ. Inf. Technol., 29: 11483-11515, 2024.

[30] F. H. Shubho et al., "ChatGPT-guided Semantics for Zero-shot Learning," in Proc. 2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA): 418-425, 2023.

Biographies

Leila Hafezi is a Ph.D. candidate in Software Engineering at Yazd University, Yazd, Iran. She received her B.Sc. degree in Computer Software Engineering from Yazd University and her M.Sc. degree in Software Engineering from Isfahan University of Technology, Isfahan, Iran. Her research interests include Natural Language Processing, Deep Learning, and Data Science.

- Email: leilahafezi@stu.yazd.ac.ir
- ORCID:0009-0009-3794-9985
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Sajjad Zarifzadeh is an Associated Professor in the Department of Software Engineering at Yazd University, Yazd, Iran. He earned his Ph.D. from University of Tehran. His research interests span Big Data Engineering, Web Mining, Internet Services and Data Analysis.

- Email: szarifzadeh@yazd.ac.ir
- ORCID: 0000-0001-5627-0542
- Web of Science researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://pws.yazd.ac.ir/zarifzadeh/en

Mohammad Reza Pajoohan is an Associated Professor in the Department of Software Engineering at Yazd University, Yazd, Iran. He earned his Ph.D. from Universiti Sains Malaysia. His research interests include Data Mining and Knowledge Discovery, Medical Data Processing, Databases and Big Data, Privacy Preservation Data Publication and Business Process Reengineering.

- Email: pajoohan@yazd.ac.ir
- ORCID: 0000-0002-7848-2025
- Web of Science researcher ID: NA
- Scopus Author ID: NA
- Homepage: https://pws.yazd.ac.ir/pajoohan/

How to cite this paper:
L. Hafezi, S. Zarifzadeh, M. R. Pajoohan, "Enhancing multi-entity detection and sentiment analysis in financial texts with hierarchical attention networks," J. Electr. Comput. Eng. Innovations, 13(2): 403-416, 2025.

DOI: 10.22061/jecei.2025.11294.790

URL: https://jecei.sru.ac.ir/article_2302.html